BUILDING ENERGY OPTIMISATION USING MACHINE LEARNING AND METAHEURISTIC ALGORITHMS

Keivan Bamdad Masouleh
Master of Engineering

Submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

School of Chemistry, Physics and Mechanical Engineering

Science and Engineering Faculty

Queensland University of Technology

2018

Keywords

Active learning methods, ant colony optimisation, artificial neural networks, building energy optimisation, building energy simulation, metaheuristic algorithms, particle swarm optimisation, robust design, sample selection method, simulation-based optimisation method, surrogate-based optimisation method, uncertainty analysis

Abstract

Buildings consume approximately 40% of end-use energy worldwide and are responsible for approximately one-third of greenhouse gas (GHG) emissions. Clearly, designing high energy performance buildings and identifying effective energy retrofit measures not only decrease CO_2 emissions, but also reduce the need for non-renewable energy sources.

While the traditional rules of thumb and building codes improve the building energy efficiency, they are likely to be far from the optimal design as they do not consider the interactions among design variables. Therefore, new methods should be developed to achieve the maximum energy savings. Building energy optimisation (BEO) is a method that considers interactions among design variables and selects the optimal building design from a set of available alternative designs based on the mathematics. A challenge of currently-available optimisation methods is that they suffer from high computational cost due to high complexities in building optimisation problems including multi-modal and nonlinear behaviour of building thermal performance, discontinuities in the optimisation variables (e.g. window type), uncertainty in building design parameters (e.g. alterations in building operating conditions) and discontinuities in the output of building simulation software (e.g. EnergyPlus). This high computational cost remains a key barrier to the widespread utilisation of optimisation as a design tool.

Accordingly, the focus of this research is on developing new efficient solution methods for Building Optimisation Problems (BOPs) and deploying them on realistic case studies to evaluate their performance and utility. Generally, BOPs can be categorised into two main groups: simulation-based optimisation (software-in-the-loop method) and surrogate-based optimisation methods. In this thesis, new methods were developed to improve the performance of both methods. Furthermore, a new methodology was developed to address uncertainty of building simulation inputs during the optimisation process to select a robust optimised design.

For the simulation-based optimisation method, two optimisation algorithms called Ant Colony Optimisation for continuous variable (ACOR), and Ant Colony Optimisation for mixed variable (ACOMV-M) for BOPs were developed for BOPs. Results demonstrated that both algorithms are noticeably more efficient than benchmark algorithms in terms of optimality, consistency, and computational cost.

For the surrogate model-based optimisation method, a new method called Surrogate model-based Optimisation using Active Learning (SOAL) was developed using active learning methods and optimisation of multiple surrogate models. Results demonstrated that proposed optimisation methods could significantly improve the performance of the surrogate-based optimisation method. Importantly, in single objective optimisation problems, the SOAL method is competitive with the simulation-based optimisation method using ACOR, with better performance in the early stages of optimisation.

To address the uncertainty of building simulation inputs during the optimisation process, a new methodology was developed based on a multi-objective scenario-based optimisation. Results demonstrated the capability of the proposed uncertainty methodology to find a robust design.

The capability of all proposed methods has been investigated by applying them to buildings Type A and Type B, recommended by Australian Building Codes Board as representatives of typical commercial buildings in Australia.

The findings of this research are significant as the proposed optimisation methods have considerably facilitated solving of BOPs. They are expected to assist building designers in meeting efnergy efficiency requirements in building codes. Moreover, applying proposed optimisation methods to buildings in different Australian climates can explore maximum potential energy savings, identify the optimal values of design variables and provide building designers with more efficient methods for designing robust energy-optimised buildings in each climate zone.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ABCB	Australian Buildings Codes Board
ACOR	Ant Colony Optimisation for continuous variable
ACOMV	Ant Colony Optimisation for Mixed Variable
ACOMV-M	Modified Ant Colony Optimisation for Mixed variable
ANN	Artificial Neural Network
BOPs	Building Optimisation Problems
DF	Derivative-Free
GA	Genetic Algorithm
MLP	Multi-Layer Perceptron
HPC	High Performance Computing
MSE	Mean Squared Error
NABERS	Australian Built Environment Rating System
NEPP	National Energy Productivity Plan
NM	Nelder and Mead
NSGA	Non-dominated Sorting Genetic Algorithm
QBC	Query By Committee
SOAL	Surrogate-based Optimisation using Active Learning
WSM	Weighted Sum Method

List of Publications

Journal papers

- Keivan Bamdad, Michael E. Cholette, Lisa Guan, John Bell, Ant colony algorithm for building energy optimisation problems and comparison with benchmark algorithms, Energy and Buildings, Volume 154, 2017, Pages 404-414.
- Keivan Bamdad, Michael E. Cholette, Lisa Guan, John Bell, Building energy optimisation under uncertainty using ACOMV algorithm, Energy and Buildings, Volume 167,2018, Pages 322-333.
- Keivan Bamdad, Michael E. Cholette, John Bell, Building Energy Optimisation Using Artificial Neural Network and Active Learning. Will be submitted soon.

Conference papers

- Keivan Bamdad , Michael E. Cholette , Lisa Guan, John Bell, Building Energy Retrofits using Ant Colony Optimisation, Healthy Buildings 2017 Europe, July, 2017, Lublin, Poland
- Keivan Bamdad, Michael E. Cholette, Lisa Guan, John Bell, Building Energy Optimisation Using Artificial Neural Network and Ant Colony Optimisation, AIRAH and IBPSA's Australasian Building Simulation 2017 Conference, Melbourne, November 2017.

Statement of Original Authorship

The work contained in this thesis has not been previously submitted to meet requirements for an award at this or any other higher education institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made.

ACKNOWLEDGEMENTS

The accomplishment of this dissertation would not have been possible without the great help and encouragement of many people. I would like to take this opportunity to thank those who have supported me during this challenging journey, which has been a unique experience in my life.

First and foremost, I would like to express my special thanks to my principal supervisor, Dr Michael Cholette, for his professional guidance, support and encouragement, and for all time and effort that he has contributed to my PhD research. It has been a precious gift to work so closely with him.

I would like to thank my associate supervisor, Professor John Bell, for his generous support and encouragement during my PhD study. He has always provided me with valuable comments on my research, and future career. I would also like to thank Dr Lisa Guan for her professional guidance and constructive feedback.

I am deeply grateful to my friends: Hamed and Mahnoosh for their great help and ongoing support since my first day in Australia.

I would like to express my special thanks to Ms Karyn Gonano for her help and valuable guidance during my PhD.

Professional editor, Diane Kolomeitz, provided copyediting services, according to the guidelines laid out in the university-endorsed ‘national guidelines for editing research theses.

I acknowledge the Science and Engineering Faculty and Queensland University of Technology for their generous financial support. I also acknowledge the Australian Building Codes Board (ABCB) for the student research scholarship.

Finally, I would like to express my deepest gratitude and heartfelt thanks to my beloved family: my mother, Simin, and my sister, Taravat, for their love,

encouragement and support over the years. I can’t express just how grateful I am, and I dedicate this thesis to them.

Chapter 1: Introduction

1.1 Background

Reducing energy consumption is one of the world's most challenging issues, particularly with increases in population and economic growth. According to the United Nations Environment Program, buildings consume approximately 40% of the world's energy and they are responsible for approximately one-third of greenhouse gas emissions in the world [1]. If no measures are taken to reduce buildings' energy consumption, GHG emissions from buildings will be almost double by 2030 [1].

In Australia, the Council of Australian Governments' (COAG) Energy Council has set a National Energy Productivity Plan (NEPP) [2], which aims to improve Australia's energy productivity by 40% by 2030, and improving the energy efficiency of buildings has been announced as one of the key measures to reach Australia's energy productivity target. In Australia, the building sector accounts for approximately 20% of total final energy consumption; it was found that energy efficiency requirements in building codes are "out of date with recent technologies" and require changes to achieve better energy efficiency outcomes [3].

Currently-used methods to design low-energy buildings are frequently based on computer simulation and *Parametric (or Sensitivity) Analysis*. In this method, in order to find values of design variables to reduce energy consumption, a base building model is first created and the design variables are varied one at a time (e.g. window-to-wall ratio) while holding others constant. This method requires a large number of building simulations, which might be impractical for all parameters. Yet, the main limitation is that this method neglects the considerable interaction between parameters. Therefore, some potential energy saving measures are either not explored, or are at suboptimal values. For example, in a building with a daylighting system, the optimised values of window and shading sizes can hardly be estimated since the use of natural light reduces energy use of artificial lighting and HVAC system (i.e. heat generation of lights) while increases solar heat gains simultaneously. Considering more variables (e.g. building orientation), makes the design problem highly complex for the maximum energy saving estimation.

With more stringent energy performance requirements and high demand for low-energy buildings, improved methods are required to design buildings to achieve maximum potential energy savings. This requires considering a combination of design parameters in the design process simultaneously, rather than merely one parameter each time.

Building Optimisation Problems (BOPs) provide a more rigorous framework for exploring new designs that manage complex trade-offs in ways that are not possible when using traditional methods. Methods for solving BOPs are primarily software-in-the-loop methods (coupling building simulation software with a mathematical optimisation algorithm). These methods seek to find the near-optimal design by intelligently exploring the candidate design values to find promising solutions and evaluating their suitability using building simulations. The extensive body of research in this area has clearly demonstrated that optimisation can dramatically reduce the energy consumption of buildings [4-14].

1.2 Research Problem

Building Optimisation Problems can be categorised into two main groups based on the method applied for optimisation [15]: simulation-based optimisation (also known as software-in-the-loop) and surrogate model-based optimisation methods.

Simulation-based optimisation (coupling building simulation software with a mathematical optimisation algorithm) is the most common building optimisation problem method and it has been applied in many studies. However, there are still a number of challenges in solving BOPs, which need to be addressed.

First, commonly used simulation-based optimisation algorithms (e.g. Genetic Algorithms and Particle Swarm Optimisation) use stochastic search strategies that require hundreds to thousands of time-consuming building simulations to converge. Optimisation time depends on many parameters such as number of objective functions and optimisation variables, and optimisation algorithm. With current computing power, some optimisation runs may take several weeks or months [16, 17]. Furthermore, as the behaviour of building thermal performance is nonlinear, the optimisation algorithm may become trapped in local minima [15]. A common strategy to avoid local minima is to restart the optimisation procedure with different initialisations, thus further increasing the computational cost [18].

Secondly, if the optimisation problem involves multiple objectives (e.g. energy consumption, thermal comfort and cost) or uncertain parameters, the number of required building simulations to find Pareto-optimal and/or robust solutions will increase significantly, which may make simulation-based optimisation methods impractical [19, 20].

This high computational cost remains a key barrier to the widespread utilisation of optimisation as a design tool [15, 17, 21]. New optimisation algorithms for BOPs could offer a solution to reduce computational cost and burden associated with the simulation-based optimisation method, which is one of the objectives in this research. Although new optimisation algorithms could improve the performance of this method, for high dimensional optimisation problems with computationally expensive building models, the simulation-based optimisation method may become computationally intractable [15, 19, 22, 23], even after applying new optimisation algorithms. Accordingly, it is necessary to develop an optimisation method that has the ability to address these computational challenges. Building energy optimisation using surrogate models (surrogate-based optimisation) is a promising method that has shown potential to find a near-optimal design in a reasonable time [19]. However, the limited number of studies conducted so far [15] have not explored how to construct surrogate models efficiently, nor fully exploited their advantages in enabling optimisation improvements.

1.3 Research Gap

The research gaps are summarised below.

- The application of both simulation-based optimisation and surrogate-based optimisation methods in buildings remains an active research area. However, both methods suffer from high computational cost to find near optimal solutions.
- Many optimisation algorithms have been developed in other fields (particularly computer science), which have shown better performance than state-of-art optimisation algorithms for BOPs in benchmark optimisation problems. However, their performance in BOPS has not been evaluated.
- A few studies considered uncertainty of building parameters (e.g. occupant behaviour) in BOPs. Current methods to address uncertainty are very time-consuming and often require probabilistic distributions of parameters, which may not be available or representative.

- The limited number of studies conducted so far [15] have not explored how to construct surrogate models efficiently nor fully exploited their advantages in enabling optimisation improvements.
- No systematic study has been conducted to compare the quality of solutions and the computational performance between surrogate-based optimisation and a simulation-based optimisation method.
- There is no study conducted to identify the optimal design of commercial buildings in Australia and explore the maximum achievable energy savings.

1.4 Research Questions

The focus of this research is on developing new efficient methods for BOPs and deploying these algorithms on realistic case studies to evaluate their performance and utility. The research gaps suggest the following questions that will be examined in this research:

1) Simulation-based optimisation. How can a new algorithm be developed to improve the simulation-based optimisation method in terms of optimality, consistency (reliably achieving solutions close to the optimal), and computational cost (number of simulations)?

A large number of optimisation algorithms have been deployed on BOPs, but a number of new optimisation approaches have been developed since most of the recent benchmarking studies. Thus, an investigation into adapting and applying these approaches to BOPs is warranted.

2) Influence of uncertainty. What is the effect of uncertainty on the "optimal" building? How can BOPs that mitigate the influence of uncertainty be formulated and solved efficiently?

While uncertainty has been investigated in a number of studies in building simulation (i.e. the “performance gap” noted in many studies), its influence on the energy consumption of the “optimal” building has not been investigated. This uncertainty is particularly important given the presence of highly uncertain parameters that depend on usage (e.g. internal loads). The impact of erroneous assumptions and methods to mitigate their influence will be studied in this research.

3) Surrogate model optimisation. How can artificial intelligence and approximation algorithms yield a more effective approach for building optimisation in terms of optimality, consistency, and computational cost?

Recent investigations into surrogate approaches have shown that they are a promising methodology for reducing the computational burden of solving BOPs. However, these preliminary approaches only used the surrogate model at the most superficial level. In particular, these studies did not investigate using the properties of the model to improve the optimisation (e.g. coupling the optimisation and the surrogate model training sample selection).

1.5 Research Aim and Objectives

This research aims to develop new optimisation methods for BOPs, which enable more widespread practical use as a building design tool. To this end, the objectives of the research are:

1) Create a new optimisation algorithm that is able to consistently find higher quality solutions with less computational cost than existing methods.

2) Deploy the new algorithm to evaluate and mitigate the influence of uncertain building simulation parameters on the resulting optimised building.

3) Develop an optimisation method, based on surrogate models, that improves the speed and/or quality of the optimised building.

1.6 Significance and Contributions

The first contribution of this research is to improve performance of simulation-based optimisation methods for BOPs. This was accomplished by development of two optimisation algorithms: ACOR algorithm for BOPs with continuous variables, and ACOMV-M algorithm for BOPs with mixed variables. Both algorithms are more efficient than current building optimisation algorithms in terms of optimality, consistency, and computational cost.

A second contribution of this research is to develop a new methodology for surrogate model-based optimisation methods. This was accomplished by the development of a new sample selection method to intelligently select samples for the surrogate model construction and development of a new method based on a committee of surrogate models in the optimisation loop.

A third contribution of this research is the development of a new methodology to address uncertainty of building simulation inputs during the optimisation process and select a robust design. This was accomplished by development of a multi-objective scenario-based optimisation and solved by the proposed optimisation algorithm (ACOMV-M).

The findings of this research are significant as the proposed optimisation method and algorithms have considerably facilitated the solving of BOPs. They are expected to aid building designers in meeting energy efficiency requirements in building codes. Moreover, applying proposed optimisation methods to buildings in different Australian climates can explore maximum potential energy savings, identify the optimal values of design variables and provide building designers with more efficient methods for designing robust energy-optimised buildings in each climate zone.

1.7 Thesis Outline

The remainder of this thesis is organised as follows:

Chapter 2 presents a comprehensive literature review. This chapter is divided into three main sections. The first section discusses simulation-based optimisation methods and reviews the current optimisation algorithms for BOPs. The second section presents a review of the application of surrogate models for both building energy prediction and optimisation. In this section, sample selection methods in other research areas (mainly computer science) are also reviewed. The third section provides a review on uncertainty in BOPs. This chapter ends by critically evaluating the literature and identifying the shortcomings and limitations of existing studies.

Chapter 3 discusses building simulation. This chapter begins with an introduction of building simulation software and subsequently details the two different buildings used as case studies in this research. In the final section, results of simulation and validation are presented.

Chapter 4 discusses a simulation-based optimisation method. This chapter is divided into five main parts. The first part introduces the optimisation problem while the second part discusses the development of simulation optimisation platform. The next part describes a new optimisation algorithm (ACOR) for BOPs and the identification of benchmark algorithms from literature. The chapter ends by presenting the optimisation results of ACOR and benchmark algorithms, followed by the conclusions.

Chapter 5 begins with the development of a new optimisation algorithm (called ACOMV-M) for solving BOPs with both continuous and discrete variables. Then, the performance of ACOMV-M was evaluated and compared against benchmark algorithms identified from literature. The final part of this chapter presents the conclusions.

In Chapter 6, optimisation under uncertainty is discussed. First, the sensitivity of optimal building parameters to three different sets of building simulation parameters (e.g. lighting loads) is investigated. Then, a multi-objective problem was developed and solved using the ACOMV-M to examine the uncertainty of building simulation parameters during the optimisation process and select a robust design. This chapter ends by presenting the results, followed by conclusion.

In Chapter 7, a new surrogate model-based optimisation method is developed and its results are compared to conventional surrogate model-based optimisation and simulation-based optimisation methods. This chapter is divided into four main sections. The first section discusses the artificial neural networks that are to serve as the surrogate model. Next, a new sample selection method is detailed, which is used in the new surrogate-based optimisation method discussed in the following section. The final section presents the results and conclusion.

Finally, Chapter 8 presents the major conclusions of the research, its limitations and recommendations for future work.

Chapter 2: Literature Review

2.1 Overview

This chapter reviews the most relevant literature related to building optimisation methods and uncertainty in BOPs. Numerous studies have been conducted so far and their application to BOPs can be categorised into two main groups:

- Simulation-based optimisation method (software-in-the-loop method)
- Surrogate-based optimisation method

Section 2.2 reviews previous studies on simulation-based optimisation methods and Section 2.3 presents the literature review on surrogate-based optimisation methods. In Section 2.4, the literature on uncertainty in BOPs is presented. Finally, research gaps for each method are identified in Section 2.5.

2.2 Simulation-Based Optimisation

The conventional method for solving BOPs is simulation-based optimisation, where building simulation software is coupled with an optimisation algorithm (e.g. Genetic Algorithm). In these methods, building simulation software plays the role of the objective function (e.g. energy consumption, thermal comfort) and the decision variables are manipulated by an optimisation algorithm to iteratively improve the objective function.

The performance of the simulation-based optimisation depends strongly on the optimisation algorithms. Figure 2.1 indicates a classification of the most-used optimisation algorithms in BOPs, according to method of operation. In addition, Table 2.1 shows the main features of each category with examples of typical algorithms.

Optimisation algorithms can be generally classified into two categories: Gradient-based algorithms and Derivative-Free (DF) algorithms.

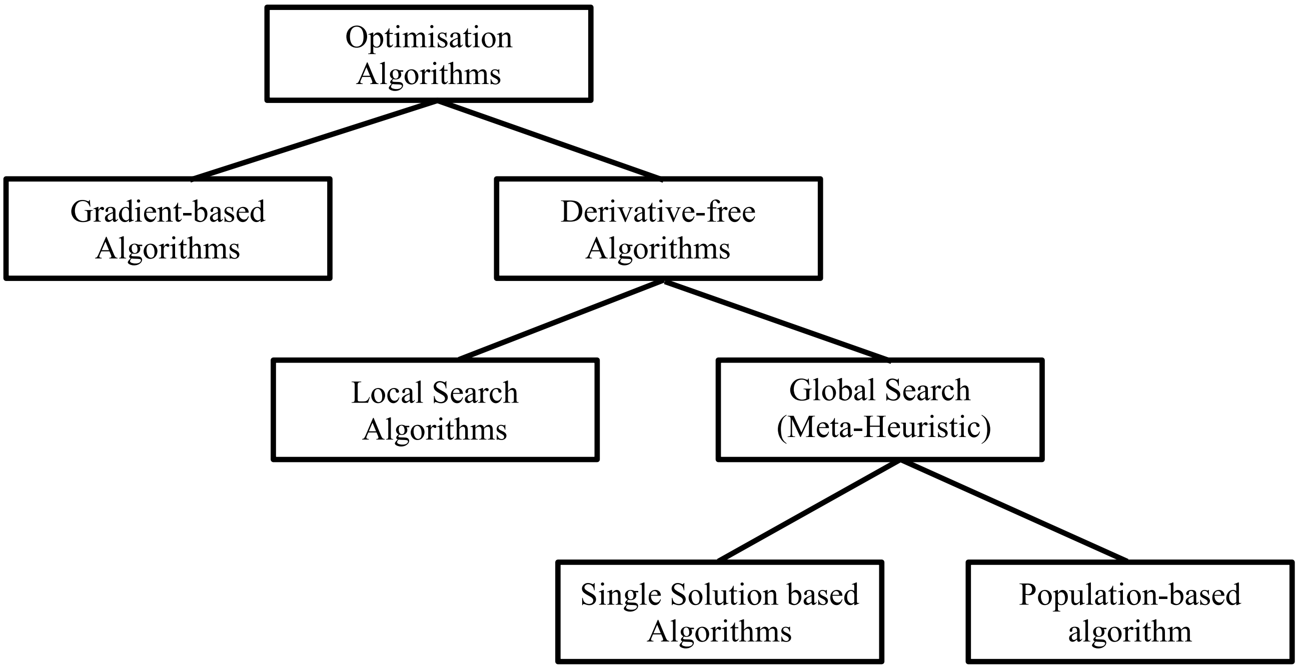
Optimisation Algorithms
Gradient-based Algorithms
Derivative-free Algorithms
Local Search Algorithms
Global Search (Meta-Heuristic)
Single Solution based Algorithms
Population-based algorithm

Figure 2.1: Classification of optimisation algorithms for BOPs

The Gradient-based methods like the Levenberg–Marquardt algorithm or Discrete Armijo algorithm use the gradient of the function to find the optimal solutions. Although these methods benefit from fast convergence and guarantee a local minimum, they are very sensitive to discontinuities in the objective functions and multi-modal functions, which cause these algorithms to be inappropriate for BOPs [15, 17, 21, 24, 25].

The second category is Derivative-Free (DF) algorithms (e.g. stochastic optimisation algorithms), which do not require the calculation of the objective function derivatives. However, these algorithms often require a large number of objective function evaluations and cannot guarantee the local optimality of the solution due to their derivative-free search mechanisms. However, the term 'optimisation' in BOPs does not necessarily mean searching for the global optima, as it may be infeasible due to the nature of either the optimisation problem or the simulation software itself [15, 24, 26]. DF algorithms are capable of dealing with both linear and nonlinear problems with discontinuities. These features make these algorithms suitable for BOPs [15, 17, 21, 27].

Table 2-1: Classification of optimisation algorithms for BOPs

Algorithms	**Features**	**Examples**
Gradient-based Algorithms	+Fast convergence +Guaranteed local optimality – Sensitive to discontinuities – May easily fall into a local minimum	Levenberg–Marquardt algorithm, Discrete Armijo Gradient algorithm
DF: Local Search Algorithms	+Less sensitive to discontinuities – May easily fall into a local minimum	Nelder-Mead Simplex, Hooke- Jeeves
DF: Global Search Algorithms	+Appropriate for nonlinear functions +Not sensitive to discontinuities – No guaranteed optimality	Evolutionary-based optimisation algorithms Swarm intelligence-based algorithms
Single Solution-based Algorithms	+Rather fast – No guaranteed optimality – May easily fall into a local minimum	Simulated annealing, Tabu search
Population-based Algorithm	+Mechanisms to avoid local optima – No Guaranteed optimality –Large number of function evaluations	GA, ACOR, PSO, Bee Colony
Hybrid Algorithms	+Combination of above features (depends on hybridisation)	PSO-HJ, CMA-ES/HDE

DF optimisation algorithms have been largely used in building optimisation studies. Peippo et al. [28] applied the Hooke and Jeeves pattern search method to identify the optimal design variables for solar energy buildings. Bouchlaghem [10] used the simplex method of Nelder and Mead and the non-random complex method to optimise the design of building envelopes. Michalek et al. [29] used Simulated Annealing (SA), Genetic Algorithm (GA) and Sequential Quadratic Programming (SQP) to search for global solutions for optimising the building design. Ant colony optimisation for discrete problems and the Radiance software were used to find a trade-off between lighting performance and cost for a media centre in Paris [30]. Wang et al. [4] applied a simulation-based optimisation method using GA to design a green building. Chantrelle et al. [31] developed a multi-criteria tool, which uses GA to optimise energy consumption, thermal comfort, cost and life-cycle environmental impact. Fesanghary et al. [5] developed a harmony search algorithm to minimise life cycle cost and carbon dioxide emissions. A graphical optimisation method was used to find the trade-off between energy and visual comfort for glazing systems in an office room [32]. Non-dominated Sorting Genetic Algorithm was used to minimise energy consumption for heating, cooling and lighting of an open space office building with respect to building envelope configurations [33]. The particle swarm optimisation algorithm (PSO) and the weighted sum method (WSM) were employed to optimise the annual cooling, heating, and lighting electricity consumption [34]. Lin et al. [35] applied Tabu Search to optimise envelope configurations for an office building.

Applying simulation-based optimisation methods frequently requires custom code development to implement an optimisation algorithm in programming language and link it to simulation software. To facilitate this process, simulation-based optimisation tools have been developed [17], which are reviewed in the next section.

Despite the many studies on BOPs, the selection of the best optimisation algorithm remains an open question, since it is highly dependent on the specifics of the problem [36]. The performance evaluation of optimisation algorithms in solving BOPs has received much attention in order to identify which algorithm performs best for BOPs.

Wetter and Wright [37] compared the performance of GA and the Hooke–Jeeves (HJ) algorithm in minimising energy consumption of a building. Their results showed that the GA has a better performance than the HJ algorithm and the latter may also fall into a local optimum more easily. Zhou et al. [38] developed an optimization module integrated with EnergyPlus and compared the performance of Nelder Mead Simplex, Quasi Newton, SA and a hybrid algorithm including GA, Tabu search and Scatter search. It was observed that Nelder Mead Simplex is the best choice for optimising a three-floor office. Mahdavi and Mahattanatawe [39] compared Hill climbing algorithm with different restart strategies with SA algorithm for maximization of preferences for temperature and visual performance, and maximization energy and visual performance preferences. It was observed that Hill climbing algorithm performed better than SA. Wetter and Wright [24] compared the performance of nine different optimisation algorithms, including a gradient-based algorithm (Discrete Armijo gradient algorithm), direct search Algorithms (Coordinate search algorithm, HJ algorithm and Simplex algorithm of Nelder-Mead), genetic algorithm, two versions of particle swarm optimisation, and Hybrid Particle Swarm Optimisation/Hooke-Jeeves (PSO-HJ) algorithm, in solving simple and complex building models. It was found that the PSO-HJ achieved the largest energy reduction among all algorithms. Their results also showed that the GA was close to the optimal point with fewer simulations than PSO-HJ. In contrast, Nelder and Mead and Discrete Armijo gradient algorithms failed to find high-quality solutions. Wright and Ajlami [40] tested the robustness of the GA in selection of control parameters in an unconstrained BOP. It was found that the GA was not sensitive to the choice of its control parameters.

More recent comparative studies have also been conducted for BOPs. Tuhus-Dubrow and Krarti [6] compared the performance of GA and PSO, and found that the GA obtained the solutions that were close to PSO, with fewer building simulations. Another study investigated the performance of GA, PSO and Sequential Search technique, and indicated that the computational efforts for the Sequential Search technique are higher than others [9]. Hamdy et al. [41] compared the performance of three multi-objective optimisation algorithms, Non-dominated Sorting Genetic Algorithm-II (NSGA-II), NSGA-II with active archive (aNSGA-II), and NSGA-II with a passive archive strategy (pNSGA-II). It was observed that aNSGA-II is more consistent in finding optimised solutions with a lower number of function evaluations than others. Hamdy et al. [42] compared the performance of seven multi-objective evolutionary algorithms with respect to different criteria. Their results indicated that two-phase optimisation using the genetic algorithm (PR_GA) can be considered the first choice for solving multi-objective BOPs. Bucking et al. [43] compared the performance of the modified Evolutionary Algorithm (EA) and Mutual Information Hybrid Evolutionary Algorithm (MIHEA) against Particle Swarm Optimisation with Inertial Weight (PSOIW) algorithm implemented in GenOpt. Results indicated that MIHEA finds better solutions with less computational time. Kämpf et al. [44] examined the performance of two hybrid algorithms called Covariance Matrix Adaptation Evolution Strategy with the Hybrid Differential Evolution (CMA-ES/HDE) and PSO-HJ in minimising the five standard benchmark functions (i.e. Ackley, Rastrigin, Rosenbrock, Sphere functions and a highly-constrained function) as well as real buildings. It was observed that the performance of CMA-ES/HDE was better than the PSO-HJ in less than ten dimensions, while if the number of dimensions exceeded ten, the PSO-HJ performed better. Another study showed that CMA-ES with sequential assessment can find the same results as GA in less time [45]. PSO showed a slightly better performance than GA in finding the optimised size for the components of solar thermal system for a single-family house [46]. Another study showed that a combination of GA with a modified simulated annealing algorithm can find more reliable results than solely the GA [47]. Recently, Futrell et al. [48] compared four optimisation algorithms in a building design for daylighting performance. They compared the Simplex Algorithms of NM, HJ, PSOIW, and PSO-HJ. They found that PSOIW found the best overall solution but PSO-HJ found solutions that are very close to the best solutions in less time.

In Australia, the application of simulation-based optimisation was conducted by Bambrook et al. [49], who applied the PSO-HJ algorithm to optimise a simple house in Sydney to design a high performance house in which cooling and heating systems were no longer needed. Table 2-2 summarizes algorithm comparative studies reviewed here.

Table 2-2: Summary of algorithm comparative studies for BOPs

Refs	Year	Algorithms	Objective function(s)	Recommended algorithm(s)	Comments
[37]	2003	GA and Hooke–Jeeves (HJ)	Energy	GA	Comparable number of function evaluations
[38]	2003	Nelder Mead Simplex, Quasi Newton, SA and a hybrid algorithm including GA, Tabu search and Scatter search	Electricity costs	Nelder Mead Simplex	Long computational cost of hybrid algorithm
[39]	2003	Hill climbing with different restart strategies and SA	Visual performance, energy and temperature	Hill climbing algorithm	
[24]	2004	Discrete Armijo gradient algorithm , Coordinate search algorithm, HJ, Nelder-Mead (NM), GA, PSO, PSO-HJ	Energy	PSO-HJ	Fast convergence of GA, unrecommended algorithms: NM and discrete Armijo
[40]	2005	GA with different parameter sets	Energy	GA	GA is insensitive to its parameters
[6]	2010	GA, PSO	life-cycle cost	PSO	GA found solutions close to PSO with fewer building simulations
[44]	2010	Covariance Matrix Adaptation Evolution Strategy with the Hybrid Differential Evolution (CMA-ES/HDE) and PSO-HJ	Energy	CMA-ES/HDE	PSO-HJ performs better for dimensions greater than 10
[9]	2011	GA, PSO and Sequential Search technique.	life cycle costs	GA, PSO	High computational time of Sequential Search approach
[46].	2012	PSO and GA	Energy and cost (solar fraction)	PSO	PSO is slightly better
[41]	2012	Non-dominated Sorting GA (NSGA-II), NSGA-II with active archive (aNSGA-II), NSGA-II with a passive archive (pNSGA-II)	Energy and life-cycle cost	aNSGA-II	better repeatability of aNSGA-II and with high convergence
[43]	2013	Modified Evolutionary Algorithm (EA), Mutual Information Hybrid Evolutionary Algorithm (MIHEA), PSO with Inertial Weight (PSOIW)	Electricity consumption	MIHEA	High convergence of both EAs
[45].	2014	CMA-ES with Sequential Assessment (CMAES-SA) and GA	Energy	CMAES-SA	Less computational time of CMAES-SA
[47]	2015	GA and hybrid GA with SA	Life-cycle cost	GA-SA	More reliability of GA-SA
[48]	2015	NM, HJ, PSOIW, and PSO-HJ.	Daylighting performance	PSOIW	Competitive algorithm: PSO-HJ
[42]	2016	pNSGA-II, two-phase optimization using the GA (PR_GA), elitist non-dominated sorting evolution strategy (ENSES), evolutionary algorithm based on the concept of epsilon dominance (evMOGA), multi-objective particle swarm optimization (MOPSO), differential evolution algorithm (spMODE-II), and dragonfly algorithm (MODA)	Energy and life-cycle cost	PR_GA	Competitive algorithms: pNSGA-II, evMOGA and spMODE-II Uncompetitive algorithms: ENSES, MOPSO and MODA

2.2.1 Building optimization tools

In this section, common optimisation tools which are mainly customised for BOPs and are based on the simulation-based optimisation method are reviewed and their main features are detailed.

2.2.1.1 GenOpt

This software was developed by Lawrence Berkeley National Laboratory and is a generic optimisation program that can be coupled with building simulation programs with input and output text files, such as TRNSYS, DOE-2, and EnergyPlus. The library of GenOpt contains different optimisation algorithms including the Golden Section and Fibonacci algorithms, the Discrete Armijo Gradient algorithm, the Nelder and Mead's Simplex algorithm, the Hooke–Jeeves, Coordinate Search, Particle Swarm Optimisation (PSO), and a hybrid PSO with the Hooke–Jeeves algorithm [50]. A drawback of the current version of GenOpt is that it does contain any multi-objective optimization algorithms. GenOpt, has been used in many studies [51-57].

2.2.1.2 BEopt

This tool which was developed by National Renewable Energy Laboratory (NREL) uses EnergyPlus simulation engine to identify optimised building design. This tool has graphical interface which allows users to select predefined options in different categories. Various discrete variables in BEopt reflect realistic construction materials and practices. Simulation assumptions in the library of BEopt are based on the building America housing simulation protocols. This tool has been used by NREL researchers and others such as [58, 59]. Limited number of predefined building options is the limitation of this tool.

2.2.1.3 MOBO

MOBO is an optimisation tool which can handle both single and multi-objective problems with continuous and discrete variables. This tool can be used with several building simulation software programs through text files such as IDA-ICE and TRNSYS. Mobo has a library of different types of optimisation algorithms such as NSGA-II, Hooke-Jeeves, Brute-Force and Random Search algorithms [60].

2.2.1.4 jEPlus:

jEPlus is a tool which is able to manage and run large and complex parametric simulations using EnergyPlus software. This tool can be coupled with optimisation algorithms to work on different types of optimisation problems. jEPlus has been used in many studies as the parametric simulation tool [61, 62] and with an optimisation algorithm for BOPs [34].

An interview conducted among 28 international building optimization experts to select an optimisation tool for BOPs. It was found that GenOpt is mostly-used tools in BOPs [16, 17].

2.3 Surrogate Based Optimisation Methods

In many engineering applications, in spite of advances in computer capacity and speed, the high computational cost remains a key issue for design and optimisation [63, 64]. To relieve the computational burden, surrogate models, also known as Meta models, are commonly used. A surrogate model is a mathematical approximation of a system, which is created using data collected by simulations or experiments to describe the behaviour of the original system. There are a lot of methods used to construct a surrogate model of a system, such as Kriging, Artificial Neural Networks (ANN), Radial Basis Function (RBF), and Support Vector Regression (SVR) [65-67].

Surrogate models have been widely used in the building science for different purposes such as design stage and operation phase (e.g. energy prediction and energy labelling) [20, 68-76]. For example, Neto and Fiorelli [74] compared the results of the neural network method and EnergyPlus with measured energy consumption. It was observed that both models are suitable for energy consumption forecast, but the neural network model is slightly more accurate than EnergyPlus. The major source of uncertainties in EnergyPlus predictions are related to lighting, equipment and occupancy schedules. Melo et al. [77] tested six different methods to generate surrogate models for building energy labelling, including multiple linear regression, multivariate adaptive regression splines, the Gaussian process, random forests, support vector machines and artificial neural networks. Results showed that the surrogate model generated by ANN has the best performance. It was also found that training time in SVR is almost six times more than ANN.

However, the application of surrogate models in BOPs is largely unexplored. Romero et al. applied a numerical method using a finite volume method to calculate energy equations and used ANN, GA and SA to optimise building design parameters [78]. Magnier and Haghighat [19] used the integration of an ANN and NSGA-II to optimise building energy consumption and thermal comfort. The average relative errors of ANN prediction were obtained around 0.5% and 3.9% for the total energy consumption and PMV, respectively. They stated that the optimisation process took approximately three weeks, while if direct coupling between simulation software and GA was used, it would require ten years to complete the task. Bianchi [79] used the ANN and GA to optimise of building energy, thermal and visual comfort. Tresidder et al. [80] compared the optimisation results of building CO_2 emissions using a Kriging surrogate model and the stand-alone GA. They found that optimisation using surrogate models leads to finding more reliable optimal solutions with fewer sampling points. They also examined both the Kriging surrogate model and the stand-alone GA on multi-objective optimisation problems with discrete variables [81]. Their results indicated that the use of the Kriging surrogate model results in a significantly better approximation of the Pareto front if the number of simulations is limited. However, they also mentioned that more investigations are required to make this conclusion robust. Eisenhower et al. [82] applied the SVR to generate the surrogate models and then compared optimisation results with the results of software-in-the-loop. They concluded that the results of these methods are approximately equivalent (in terms of numerical quality). Gossard [83] used the ANN and NSGA-II to optimise the annual energy consumption and the summer comfort degree index in a building for two French climates. Gengembre et al. [84] optimised the life cycle cost of a single-zone building using a Kriging surrogate model. The results indicated that acceptable accuracy was achieved by the Kriging model at the reasonable computational time. Asadi [85] applied GA and ANN for the optimisation of the three objective functions: energy consumption, retrofit cost, and thermal discomfort hours, in a school building.

2.3.1 Sample Selection Methods

The performance of the surrogate models depends strongly on the number and quality of sample points collected by experiments or computer simulation. More sample points provide the surrogate model with more information, and consequently this leads to more accurate predictions but at a higher computational cost [64, 86]. Hence, the main challenge of constructing surrogate models is to achieve the highest prediction accuracy with the least computational cost. The process of determining the samples that will be used to estimate the surrogate is called *Sample Selection*.

The most widely used sample selection method is random sampling. In this method, sample points are randomly selected to train the surrogate model. Due to the random selection, some samples may contain less information and not be representative of the whole design space. Therefore, more sample points (and higher computational cost) should be added to the training dataset to construct a surrogate model with desired prediction accuracy.

To address this trade-off, *Active Learning* methods have been developed with the aim of evaluating the "informative-ness" of the unlabelled samples and selecting the most informative samples through different query strategies. These strategies could be classified into six groups [87]: uncertainty sampling, query by committee (QBC), expected model change (expected gradient length), expected error reduction, variance reduction, and density weighted methods. Most studies that applied sample selection methods are in the context of classification problems [87], while a few studies used them in regression problems (i.e. function approximation problems).

Krogh and Vedelsby [88] defined the ambiguity as the variation of the output of ensembles of neural networks over unlabeled data. They used the ambiguity to select new training data and reduce the generalisation error for the square-wave function. RayChaudhuri and Hamey [89] used ensembles of neural networks similar to [88] to reduce the generalisation error. However, they used random subsamples of a small amount of data to train the ensemble of neural networks. Burbidge et al. [90] investigated the performance of the committee-based approach for active learning in the one dimensional mathematical problem. Their experience showed that this approach only works when the model class is correctly specified and data are noise free. Cohn et al. [91] proposed a statistical active learning method and computed the approximation of variance for Gaussian mixture models, neural networks, and locally-weighted linear regression to reduce the generalisation error in the “Arm2D" problem. Yu et al. [92] proposed a passive sampling method based on geometric characteristics of data points in the feature space. They found that passive sampling outperformed random sampling and active learning based on predicted regression error for noisy datasets while active learning performed best for noiseless datasets. Douak et al. [93] developed three different active learning strategies for kernel ridge regression based on pool of repressors, Euclidean distance and residual regression for minimisation of prediction error for wind speed. Results showed that a smart collection of samples could improve the model’s prediction for wind speed problems. Zhao et al. [94] developed ANN and SVM models for wind speed forecast with an active learning approach based on selecting samples with the higher Euclidean distance and lower cross-validation error. Results indicated that their proposed method could significantly reduce the number of training samples and ensure model accuracy. Recently, Verrelst et al. [95] applied different active learning methods to the biophysical variable retrieval problem. They compared random sampling with six active learning methods, including a variance-based pool of regressors, entropy query by bagging [96], residual regression active learning [93], Euclidean distance-based diversity, angle-based diversity [97], and cluster-based diversity [98]. Results showed all active learning methods outperformed random sampling.

2.4 Uncertainty in Building Optimisation Problems

In the vast majority of simulation and optimisation problems, building designers assume that building input parameters are deterministic (or perfectly known). However, in real building problems, especially at the early stages of building design, parameters are often highly uncertain. These uncertainties may arise from different sources, including uncertainties in the thermophysical properties of construction materials and in weather data, lack of designers' knowledge of building occupancy, occupant behaviour and appliance loads, and uncontrolled infiltration rates [99, 100]. These uncertainties cause a significant discrepancy between the predicted and actual building energy performance [101-103]. In Building Performance Simulation (BPS), the impact of uncertainty in building simulation assumptions has been investigated by a number of studies [99, 104-111]. For example, Silva [109] analysed the uncertainties in occupant behaviour and physical parameters for a residential building simulated by EnergyPlus software and found up to a 43.5% deviation in energy consumption.

In contrast to BPS problems, studies considering uncertainty in BOPs are quite limited. Hoes et al. [112] proposed a building performance indicator based on uncertainty in the users' behaviour to rank Pareto solutions to select the most robust solution. They used Monte Carlo Simulation and NSGA-II to calculate and minimise the mean value of building performance indicators. Bucking [113] applied Monte Carlo Simulation and an evolutionary algorithm to optimise energy consumption and life-cycle cost under economic uncertainty. To address the well-known issue of high computational cost for Monte Carlo Simulation, Ramallo-González et al. developed a Changing Environment Evolutionary Strategy (CEES) to optimise energy under uncertainty in occupant behaviour [114]. In this strategy, the algorithm's populations are evaluated with a different environmental parameter at each generation. In another study, Hopfe et al. [100] developed a Kriging meta-model of building performance and used Monte Carlo Simulation to do optimisation under uncertainty. However, construction of a sufficiently accurate meta-model is a key factor in the performance of the overall surrogate-based optimisation problems (which was not discussed in [100]). This construction depends strongly on the samples that are used in training the meta-model and the selection of free parameters, which have no generally accepted guidelines for their selection and require significant expertise and/or time to properly tune [115].

In addition to the issue of high computational cost, probability models (e.g. for Monte Carlo simulation) require probabilistic distributions of parameters that may not be available, particularly in light of the fact that uncertainties may change during the building life time [100]. In such cases, *scenario analysis* (i.e. analysing the behaviour of the building under a number of different specific building assumptions) may provide a complementary tool to enable uncertainty analysis when detailed distributional information is lacking [116]. Recently, Kotireddy et al. [117] applied scenario analysis and the minimax regret method as the measure of performance robustness to identify robust designs. The preferred robust design is selected based on performance robustness and optimal performance.

2.5 Summary

The application of both simulation-based optimisation and surrogate-based optimisation methods in buildings remains an active research area. However, both methods suffer from key issues to find optimal (or near optimal) solutions. Table 2-2 shows the potential improvments for each method identified from the literature.

- In simulation-based optimisation, the performance of the method depends strongly on the optimisation algorithm. This method requires hundreds to thousands of time-consuming building simulations to find near-optimal solutions. This high computational cost is likely the key reason why this approach remains impractical in the building industry. Different optimisation algorithms were applied to improve the performance of simulation-based optimisation methods in terms of optimality and reducing computational cost. Comparative studies indicated that Particle Swarm Optimisation with Inertia

Weight (PSOIW) and the hybrid PSO-HJ algorithms perform well on BOPs [43, 44, 46, 48], outperforming many other popular optimisation algorithms (e.g. GA).

- Surrogate-based optimisation is promising, especially for optimisation of computationally expensive models [15]. However, the limited number of studies conducted so far have not explored how to construct surrogate models efficiently, and nor have they fully exploited their advantages in enabling optimisation improvements.
- No systematic study has been conducted to quantify and evaluate the computational performance gains that may be expected using a surrogate approach.
- ANNs are the most used surrogate models for both building energy prediction and optimisation problems, including many building studies [19, 20, 68-72, 77, 79, 83, 85]. The performance of the surrogate method depends strongly on the number and quality of samples used to create the surrogate model. All studies for BOPs used the random sampling method, which suffers from extra computational cost for labelling non-informative samples.
- For regression problems (surrogate models) the literature in other fields (particularly computer science) revealed that active learning methods based on *"Query By Committee"* (QBC) have shown promise to improve the efficiency of constructing surrogate models.
- There is no study investigating the active learning methods for either building energy prediction or building optimisation problems.
- A few studies considered uncertainty of building parameters (e.g. occupant behaviour) in BOPs. Current methods to address uncertainty are very time-

consuming and often require probabilistic distributions of parameters. Coupling these methods to optimisation methods to find a robust solution causes BOPs to be computationally too expensive.

- Very few studies applied optimisation methods for building design in Australia, and the potential of these methods for energy savings has not been fully explored.

Chapter 3: Building Simulation

3.1 Overview

In this chapter, the simulation of building energy performance is detailed and the selection and validation of case studies for optimisation are discussed. Section 3.2 discusses building simulation software and Section 3.3 details two representative commercial buildings, which are used in this research as optimisation case studies. Finally, Section 3.4 presents the simulation results and validation of these building models.

3.2 Building Simulation Software

Building performance simulation tools play a key role in building design. There are several building energy simulation programs, such as EnergyPlus [118], TRNSYS [119], DesignBuilder [120] and IES-VE [121], which are widely used in industry and the scientific community due to their high capability and reliability. However, the best choice for a specific project depends on different factors, such as designer knowledge, client needs, required level of accuracy and simulation time [20].

For building optimisation problems (particularly those involving novel algorithms), building simulation software is coupled with an external program. Thus, the simulation software must have some specific features to be applicable for optimisation. The main features include (but not limited to):

- Reading and writing ASCII text input and output files
- Generating output with various types of formats

- Running simulation software in a batch process
- Allowing parallel simulation runs
- Enabling simulation of advanced HVAC systems
- Calculating different thermal comfort metrics such as Predicted Mean Vote (PMV) and ASHRAE (Standard 55)
- Simulating daylighting controls and calculating the effect of reduced artificial lighting on building loads
- Being compatible with both Windows and Linux, which is necessary for high performance computing

EnergyPlus, developed by the US Department of Energy, is whole building energy simulation software, which benefits from all aforesaid features. This software was selected as the first choice among building optimisation experts for BOPs [17]. Accordingly, EnergyPlus was selected as the building simulation software in this research.

3.3 Building Modelling: Case Study Description

In this research, two reference buildings called Type A and Type B, developed by the Australian Buildings Codes Board (ABCB), are used. Reference buildings aim to represent a typical building in the national building stock to ensure that results from energy analyses are representative [122]. Using reference buildings helps designers to understand how real buildings in a specific climate zone are likely to be affected by any energy saving measures. Thus, these buildings were considered as suitable case studies in this research. Moreover, ABCB reference buildings have been widely used in many studies [102, 123-133]. However, different simulation assumptions and input values have been used in the literature, which has led to different building simulation results. In this research, the building configuration, parameters, and assumptions (e.g. internal loads) are as specified in the ABCB recommendations [102, 134, 135]. The details of these buildings are discussed in the following sections.

3.3.1 Building Type A

Building Type A is an office building (10 storey tower) with heavy-weight concrete construction and a gross floor area of 9985 m^2. This building includes all features of real buildings, including multiple thermal zones, internal loads of occupancy, lighting, equipment, auxiliary service equipment and HVAC system. The template VAV system of the EnergyPlus was selected to model a variable air volume system with water-cooled chiller (COP = 3.57) and the heating and cooling sizing factors are 1.25. The prototypes and details of building Type A are given in Figure 3.1, Table 3.1 and Table 3.2. The schedules used for occupancy, lighting (limited control), equipment and HVAC working hours are the same as given by the National Australian Built Environment Rating System (NABERS) [129].

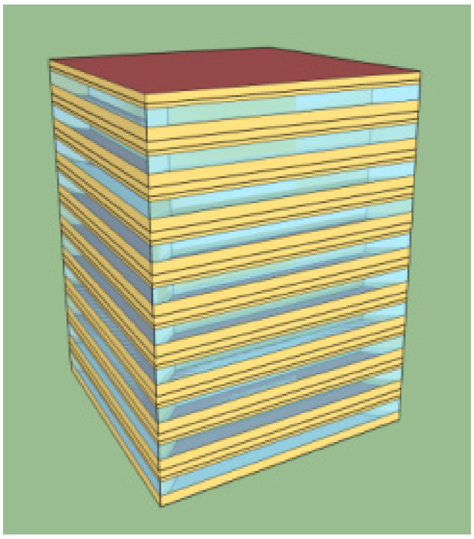

Figure 3.1: Ten-storey building Type A (ABCB) [134, 135]

Table 3-1: Building Type A construction details [134, 135]

	Construction Materials	Overall U-Value (W/m^{2}K)
Wall	200 mm heavy weight concrete R1.5 batts, 10mm plasterboard (absorption coefficient (AC) = 0.6)	0.557
Roof	Metal deck, air gap, 150mm HW concrete, roof space, R2.0 batts, 13mm acoustic tiles (AC= 0.6)	0.231
Floors	175 mm concrete, carpet 2.7 cm	1.351
Windows	6 mm clear glass (SHGC = 0.818, VT= 0.88)	5.89
Window to wall ratio	38 %	

Table 3-2: Building geometry details and assumptions used in building modelling

Parameters	Values
Total floor area (m^2)	9985.6
Geometry (m)	31.6 × 31.6
Number of floors	10
Floor to floor height (m)	3.6
Floor to ceiling height (m)	2.7
Lighting load	15 W/m^2
Equipment load	15 W/m^2
Lifts and auxiliary service equipment	1 W/m^2
Occupancy	0.1 Person/m^2
Temperature set-point	20-24 °C
Temperature set-back	28 °C (18:00-07:00, business days)
Infiltration	1 ACH outside HVAC operating hours, no infiltration during HVAC hours
HVAC system	VAV system, water cooled AC, Gas boiler, COP=3.57 (no heat recovery and economy cycle)

3.3.2 Building Type B

Building Type B is a three-storey office building with heavy-weight concrete construction and a gross floor area of 2003.85 m^2. This building includes all features of real buildings including multiple thermal zones, internal loads of occupancy, lighting, equipment, auxiliary service equipment and HVAC system. The template VAV system of the EnergyPlus was selected to model a variable air volume system with water-cooled chiller (COP = 3.57) and the heating and cooling sizing factors are 1.25. The details of building Type B are given in Figure 3.2, Table 3.3 and Table 3.4. The schedules used for occupancy, lighting (limited control), equipment and HVAC working hours were the same as given by NABERS [129].

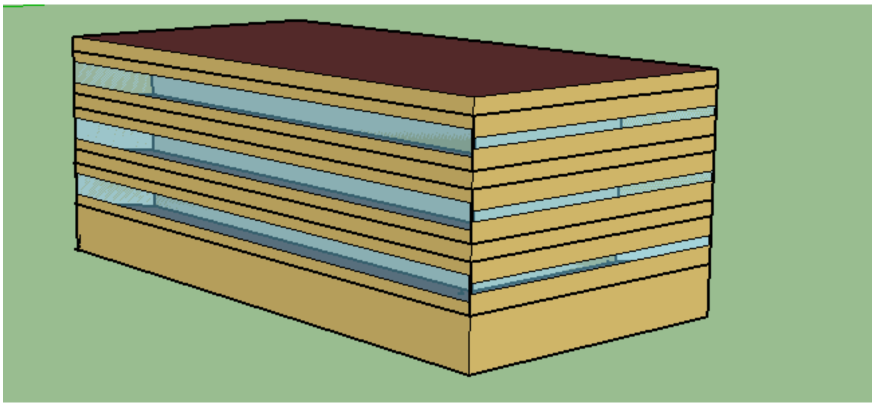

Figure 3.2: Three-storey building Type B (ABCB) [134, 135]

Table 3-3:Building construction details [134]

Component	Construction Materials	U-Value (W/m^{2}K)
Wall	200 mm heavy weight concrete R1.5 batts, 10mm plasterboard (absorption coefficient=0.6)	0.520
Roof	Metal deck, air gap, 150mm heavy weight concrete, roof space, R2.0 batts, 13mm acoustic tiles (absorption coefficient=0.6)	0.267
Floors	175 mm concrete, carpet 2.7 cm	1.351
Windows Window to wall ratio	6 mm clear glass 37.5 % (E & W faces), 15% (N & S faces)	5.89
Overhang	NA	

Table 3-4: Building geometry details and assumptions used in building simulation [134]

Parameters	Values
Total floor area (m^2)	2003.85
Geometry (m)	36.5 × 18.3
Number of floors	3
Floor to floor height (m)	3.6
Floor to ceiling height (m)	2.7
Lighting load	15 W/m^2
Equipment load	15 W/m^2
Lifts and auxiliary service equipment	1 W/m^2
Occupancy	0.1 Person/m^2
Temperature set-point	20-24 °C
Temperature set-back	28 °C (18:00-07:00, business days)
Infiltration	1 ACH outside HVAC operating hours, no infiltration during HVAC hours

3.4 Simulation Results Comparison

Buildings Type A and Type B are theoretical buildings and therefore validation against measured energy consumption data is not possible. However, since the buildings are considered to be representative of generic office buildings, the simulation results were compared to the average state energy intensity for office buildings from [102] for four cities with diverse climates: Darwin, with hot humid summers and warm winters; Brisbane, with warm humid summers and mild winters; Melbourne with warm summers and cool winters; Hobart, with mild to warm summers and cold winters [124].

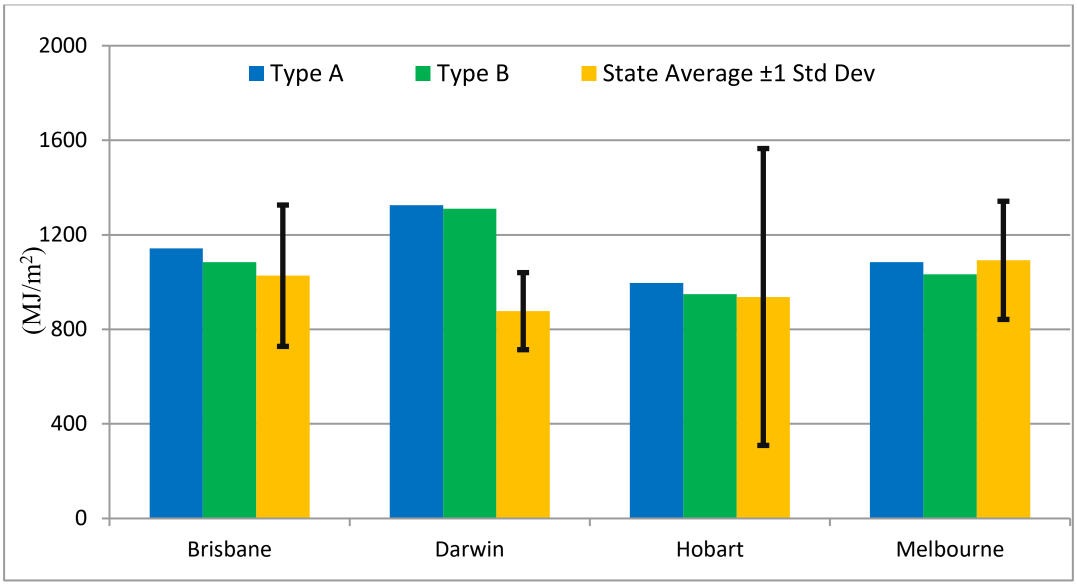

Figure 3.3: Simulation results for both buildings Type A and Type B

Figure 3.3 shows the simulation results for the annual energy consumption per unit floor area for both buildings Type A and Type B, and the average state energy intensity of office buildings. For all cities except Darwin, the simulation results of annual energy consumption are in close agreement with the corresponding state average (within one standard deviation of those reported [136]). In addition, for all cities, the simulation results of the present study are very close to the study conducted by Daly et al. [102]. The discrepancy between simulation results and state average for Darwin was also reported in [102]. Some possible reasons for this discrepancy include: different building constructions in that climate, higher cooling set-points, differences in occupant behaviour [102], different cooling COPs and higher infiltration rates or these reference buildings may not be an appropriate representative of commercial buildings for Darwin.

3.5 Summary

In this chapter, two representative commercial buildings recommended by ABCB were chosen as case studies. The simulation of building energy performance was detailed and the simulation results were compared with state average (within one standard deviation) and another study. The comparison showed that simulation results are in close agreement with both of them.

Chapter 4: Simulation-Based Optimisation with Ant Colony Optimisation

4.1 Overview

Simulation-based optimisation, also known as software-in-the-loop, is the most commonly used method for BOPs. This method consists of three phases: pre-processing, optimisation, and post-processing [15]. The pre-processing phase plays a significant role in the optimisation and mainly contains the formulation of the optimisation problem. In this phase, objective functions, problem constraints, optimisation variables, and an optimisation algorithm are determined. In addition, coupling the optimisation algorithm to building simulation software is done in this phase. In the optimisation phase, the main tasks are running the optimisation and monitoring convergence of optimisation. The post-processing is comprised of the interpretation of optimisation results.

In this chapter, Ant Colony Optimisation is adapted and applied to BOPs for the first time, and benchmarked against state-of-the-art algorithms. This algorithm is selected because applying ACOR on mathematical test functions, such as Sphere, Tablet and Rosenbrock, showed that ACOR is an competitive algorithm in the family of metaheuristic algorithms, outperforming other metaheuristic algorithms such as GA in some test functions [137]. Moreover, ACOR has illustrated high efficiency in other domains [138-141]. However, its application in building optimisation problems has not been reported to date.

This chapter is organised as follows: Section 4.2 details the problem statement and Section 4.3 details a platform to couple simulation software to an optimisation algorithm. Section 4.4 details the ant colony optimisation algorithm along with benchmark optimisation algorithms. Finally, Section 4.5 presents the optimisation results.

4.2 Problem Statement

The building optimisation problem considered here can be formally stated as

$$\begin{gathered} \min\ f(\mathbf{x}) \\ \text{subject to: } \mathbf{x} \in \mathbb{X} \subseteq \mathbb{R}^{\mathrm{N}} \end{gathered} \qquad 4.1$$

Where $f(\cdot): \mathbb{X} \rightarrow \mathbb{R}$ is the objective function, $\mathbb{X} \subset \mathbb{R}^{\mathrm{N}}$ is the feasible space, $\mathbf{x} = [\mathrm{x}_1, \mathrm{x}_2, \dots, \mathrm{x}_{\mathrm{N}}]$ is the vector of independent design variables. For the BOP considered here, the feasible design space is simply stated in terms of upper and lower bounds on parameters: $-\infty < l_i \leq x_i \leq u_i < +\infty,\ i = 1,2, \dots, N$ where ℓ_i and u_i are the lower bound and the upper bound of the variable i. Since the decision variable input ranges can be normalised, we may assume (without loss of generality) that $\ell = 0$ and $u = 1$.

According to the research conducted among an international group of building optimisation experts, energy and cost have been identified as the most used objective functions, and systems and controls, and envelope variables as the most optimised variables in BOPs. However, the selection of the variable depends on the innovation of the project and the complexity of variable. Furthermore, thermal comfort and cost were defined as the main constrains [16, 17]. In this study, the objective function, $f(\cdot)$, is the building annual end-use energy consumption ($\mathrm{MJ/m^2}$ Year), which is calculated by EnergyPlus [118], which can be written as follows:

$$f(\mathbf{x}) = E_c(\mathbf{x}) + E_f(\mathbf{x}) + E_l(\mathbf{x}) + E_p(\mathbf{x}) + E_h(\mathbf{x}) + E_m(\mathbf{x}) \quad 4.2$$

where E_c is the energy consumption for space cooling $(\mathrm{MJ/m^2\ Year})$, E_f is the energy consumption of the supply and return fans of HVAC system $(\mathrm{MJ/m^2\ Year})$, E_l is the energy consumption of lighting $(\mathrm{MJ/m^2\ Year})$, E_p is the energy consumption of pumps $(\mathrm{MJ/m^2\ Year})$, E_h is the energy consumption for space heating $(\mathrm{MJ/m^2\ Year})$ and E_m is the energy consumption of both interior equipment and heat rejection[1]$(\mathrm{MJ/m^2\ Year})$.

4.3 Development of an Automatic Optimisation Platform

Simulation-based optimisation is a complex task that frequently requires hundreds to thousands of building simulations to find the optimal (or near optimal) solution. Therefore, a platform is needed to control the whole optimisation process. Some software tools that do simulation-based optimisation (e.g. GenOpt), however, are limited to predefined optimisation algorithms. Thus, in order to evaluate the performance of new optimisation algorithms, a new simulation-optimisation platform is required, which has the maximum flexibility to implement custom optimisation algorithms. For this purpose, MATLAB was selected to generate and analyse EnergyPlus files. A script in MATLAB was developed to control the optimisation process. This script is able to read and write text files, call EnergyPlus, and evaluate the objective function. The data exchange between this script and EnergyPlus is done through text files. In each iteration during the optimisation process, the optimisation algorithm generates a new solution and calls EnergyPlus to simulate it. The MATLAB

[1] For the HVAC system considered, heat rejection is the energy consumption of cooling tower fan.

script reads the EnergyPlus output text file, extracts relevant quantities, and evaluates the values of objective functions. According to this evaluation, the optimisation algorithm creates a new solution to reduce objective function and calls EnergyPlus again. This process is repeated until the convergence criterion is met. Figure 4.1 outlines this process. This simulation-optimisation platform serves as a basis for testing and developing simulation-in-the-loop optimisation. In addition, the platform was adapted to address the sample selection problem, which is discussed in Chapter 7.

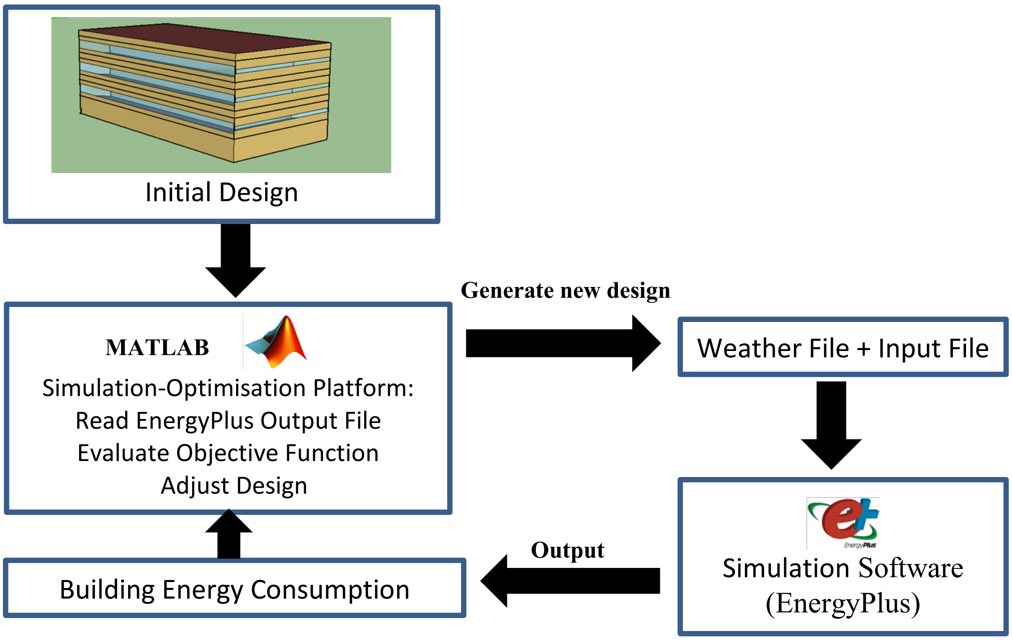
Initial Design
Generate new design
MATLAB
Simulation-Optimisation Platform:
Read EnergyPlus Output File
Evaluate Objective Function
Adjust Design
Weather File + Input File
Output
Building Energy Consumption
Simulation Software
(EnergyPlus)

Figure 4.1: Simulation- optimisation platform

4.4 Optimisation Algorithm Development

Metaheuristic optimisation algorithms are often the first choice for BOPs due to discontinuities and the nonlinear thermal behaviour of buildings [15, 17, 21]. In this research, Ant Colony Optimisation for continuous domain (ACOR) was developed and its results were compared against benchmark algorithms. The next section explains the details of this algorithm.

4.4.1 Ant Colony Optimisation Algorithm

Ant Colony Optimisation (ACO) is a metaheuristic algorithm that was inspired by observations of ant behaviour. This algorithm was first designed to solve discrete optimisation problems and later extended to continuous variables [137, 142]. This extension, called Ant Colony Optimisation for continuous domain (ACOR) [137], was employed to optimise building energy performance. A strategy to deal with boundary constraints was added to the original ACOR algorithm in this research.

ACOR operates on a *solution archive*, which is shown in Figure 4.2. This archive contains the values of the N decision variables $\mathbf{x}_j = \left[x_j^1, x_j^2, \dots, x_j^N\right]$ and the associated objective function values $f\left(\mathbf{x}_j\right)$, obtained by simulating the building to obtain the annual energy consumption. Solutions in the archive are sorted from lowest to highest objective values, i.e.

$$f(\mathbf{x}_1) \leq f(\mathbf{x}_2) \leq \dots \leq f\left(\mathbf{x}_j\right) \leq \dots \leq f(\mathbf{x}_M) \qquad 4.3$$

x_1^1	x_1^2	…	x_1^i	…	x_1^N
x_2^1	x_2^2	…	x_2^i	…	x_2^N
$\vdots$	$\vdots$	…	$\vdots$	$\vdots$	$\vdots$
x_j^1	x_j^2	…	x_j^i	…	x_j^N
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
x_M^1	x_M^2	…	x_M^i	…	x_M^N

$f(\mathbf{x}_1)$	ω_1
$f(\mathbf{x}_2)$	ω_2
$\vdots$	$\vdots$
$f(\mathbf{x}_j)$	ω_j
$\vdots$	$\vdots$
$f(\mathbf{x}_M)$	ω_M

Figure 4.2: Solution archive for ACOR (adapted from **[137]**)

New candidate solutions are generated according to a Gaussian kernel probability density function (PDF) based on the solutions in the archive

$$G^i(x) = \sum_{j=1}^{M} \omega_j \, g_j^i(x) = \sum_{j=1}^{M} \omega_j \frac{1}{\sigma_j^i \sqrt{2\pi}} \, e^{-\frac{\left(x-\mu_j^i\right)^2}{2\sigma_j^{i2}}} \qquad 4.4$$

where $G^i(x)$ is the Gaussian kernel for the ith dimension of the solution, $g_j^i(x)$ is the jth sub-Gaussian function for the ith dimension while μ_j^i and σ_j^i are the jth dimensional mean value and the standard deviation, respectively. The *weights* ω_j are set so that solutions with lower objective values are preferred, since they likely indicate neighbourhoods where good solutions may be found. The weights are assigned based on the position of a solution in the archive

$$\omega_j = \frac{1}{qM \sqrt{2\pi}} \, e^{-\frac{(j-1)^2}{2q^2M^2}} \qquad 4.5$$

where q is a free parameter of the algorithm, which controls how sharply the weights decrease with the archive index j. Low values of q increase the weights of the best solutions relative to the other solutions in the archive.

The mean and standard deviation of the sub-Gaussians are also set, based on the archive solutions

$$\mu_j^i = x_j^i \qquad 4.6$$

$$\sigma_j^i = \xi \sum_{\ell=1}^{M} \frac{\left|x_\ell^i - x_j^i\right|}{M-1} \qquad 4.7$$

In other words, the standard deviation is set according to the average distance of $\mathbf{x}_j$ from the other $M-1$ solutions in the archive along dimension i in the parameter space. The free parameter ξ is simply a scaling factor, which allows users to set the percentage of this average.

The new candidate solutions are generated according to the distribution in Eq. 4.8 via a two-stage process. First, a solution from the archive is randomly selected with probability

$$P_j = \frac{\omega_j}{\sum_{r=1}^{M} \omega_r} \qquad 4.9$$

Obviously, it is more probable that solutions with higher ω_j will be selected. A new candidate solution, $\tilde{\mathbf{x}}$ is randomly generated according to the component-wise probability density functions

$$g_j^i(\tilde{x}^i) = \frac{1}{\sigma_{\mathrm{j}}^i \sqrt{2\pi}} e^{-\frac{\left(\tilde{x}^i - \mu_{\mathrm{j}}^i\right)^2}{2\sigma_{\mathrm{j}}^{i2}}} \qquad i = 1,2,\ldots,N \qquad 4.10$$

where j is the selected solution from the archive. The objective value of this solution is then evaluated and the generation procedure repeats until m candidate solutions are generated. The archive is then updated by selecting the best M solutions from the $M + m$ solutions. To conduct the optimisation with ACOR, all variables are normalised between zero and one ($\ell_i = 0$ and $u_i = 1$). However, during the generation of new solutions, a variable (x_i) may violate the domain boundary constraint. If this occurs, x_i is repaired as follows:

$$\begin{aligned} \text{if } x_i < 0 \rightarrow \quad x_i &= |x_i| \\ \text{if } x_i > 1 \rightarrow \quad x_i &= 1 - (x_i - floor(x_i)) \end{aligned} \qquad 4.11$$

The ACOR algorithm is summarised below.

0. Select values for the parameters $q, \xi, M, m \leq M$

1. Initialise. Randomly generate $\mathbf{x}_j \ \ j = 1,2,\dots,M$ according to component-wise uniform distributions[2] between the upper and lower bounds. Compute the objective values.
2. Sort solutions in ascending order according to their objective values so that Eq. 4.3 is satisfied.
3. Calculate weights according to Eq. 4.5.
4. Generate a new solution.
    a. Select a solution j from the archive with probabilities from Eq. 4.9
    b. Generate a solution according to Eq. 4.10
    c. Adjust any variable values violating constraints according to Eq. 4.11
5. Repeat step (4) m times.
6. Evaluate objectives of m new solutions.
7. Select the best M solutions from the $M + m$ solutions available.
8. Check stopping criteria. If they are not satisfied, return to 2.

A key challenge in the application of any optimisation algorithm is striking the proper balance between exploration of the parameter space and intensification of the search near quality solutions. In ACOR this behaviour is controlled using the parameters q and ξ. Smaller values of q promote intensification by assigning relatively large weights to better solutions in the archive and thus generating more candidate solutions in the neighbourhood of the best solutions. Larger values of q increase exploration, by assigning more uniform weights to solutions in the archive. The parameter ξ is a normalised width of the sub-Gaussians, in which its higher values

[2] One could also use a space-filling algorithm (e.g. Latin Hypercube) to conduct this step.

promote increased exploration around a given solution, while its lower values increase intensification near it.

4.4.2 Benchmark Algorithms

Particle swarm-based algorithms were selected as benchmark metaheuristic algorithms, as the literature review revealed their efficiency for BOPs (see Section 2.2 for more discussion). In addition, the NM algorithm was also selected as a benchmark direct search algorithm. These algorithms are detailed in this section.

4.4.2.1 Particle Swarm Optimisation

The selected benchmark algorithms are both based on Particle Swarm Optimisation (PSO), which is inspired by the social behaviour of birds. PSO is a metaheuristic optimisation algorithm introduced in [143], which seeks the optimum solution by changing the position and velocities of "particles" (which represent particular values of the building parameters in this study).

The first benchmark algorithm will be Particle Swarm Optimisation with Inertia Weight (PSOIW), which was developed to improve the performance of the original PSO by better controlling the balance between global and local searching [144, 145]. In PSOIW, the velocity and position of a particle are determined as follows:

$$\mathbf{v}_i(k+1) = \omega(k)\mathbf{v}_i(k) + c_1\rho_1(k)(\mathbf{p}_{l,i}(k) - \mathbf{x}_i(k)) + c_2\rho_2(k)(\mathbf{p}_{g,i}(k) - \mathbf{x}_i(k)) \quad 4.12$$

$$\mathbf{x}_i(k+1) = \mathbf{x}_i(k) + \mathbf{v}_i(k+1) \quad 4.13$$

where $\mathbf{x}_i$ is the position of ith particle, k is the generation number, $\mathbf{v}_i$ is the particle velocity, ρ_1 and ρ_2 are uniformly distributed random numbers. The variable $\mathbf{p}_{l,i}(k)$ is the position of the particle with the best objective value observed so far for particle i, $\mathbf{p}_{g,i}(k)$ is the position or the particle with the best objective value so far[3], $\omega(k)$ is a non-increasing inertia weight, and c_1 and c_2 are algorithm parameters that control the relative influence of the global and local optima on the particle velocity update in Eq. 4.13.The inertia weight is computed as follows:

$$\omega(k) = \omega_0 - \frac{k}{K}\,(\omega_0 - \omega_1) \qquad 4.14$$

where $\omega_0 \in \mathbb{R}$ is the initial inertia weight, $\omega_1 \in \mathbb{R}$ is the inertia weight for the last generation $(0 \leq \omega_1 \leq \omega_0)$, and $K \in N$ is the maximum number of generations. The PSOIW algorithm is summarised below [144, 145].

PSOIW Algorithm

```
0. Select values of algorithm parameters ($c_1$ and $c_2$), number of particles $n_P \in$ N and number of generations $n_G \in$ N
1. Initialise particles ($\mathbf{x}_i$) and velocities ($\mathbf{v}_i$)
2. Evaluate the objective function values of each of the particles and determine the global best particle ($\mathbf{p}_{g,i}(k = 0)$)
3. Compute the inertia weight $\omega(k)$ (Eq. 4.14)
4. Update the particles' velocity $\{\mathbf{v}_i(k + 1)\}_{i=1}^{n_P}$
5. Update the particles' location $\{x_i(k + 1)\}_{i=1}^{n_P}$
6. For $i \in \{1, \dots, n_P\}$, determine the local best particles ($\mathbf{p}_{l,i}(k)$) and the global best particle ($\mathbf{p}_{l,i}(k)$)
```

[3] Actually, the $\mathbf{p}_{g,i}$ is the best objective found amongst the particles in a neighbourhood of particle i, which could potentially be all particles.

7. Check the stopping criterion $(k = n_G)$. If it is not satisfied, replace k by $k + 1$, and go to Step 3

4.4.2.2 Hybrid Particle Swarm Optimisation and Hooke-Jeeves Algorithm

The next benchmark algorithm is the hybrid PSO-HJ algorithm. PSO (as detailed in the previous section) searches globally to find near-optimal solutions and then Hooke-Jeeves (HJ) searches locally to refine the solutions. PSO stops in this hybrid algorithm after a finite number of iterations or generations and then Hooke-Jeeves refines the PSO solution and terminates when no improvement is found [24]. The next section details the Hooke-Jeeves algorithm.

4.4.2.2.1 Hooke-Jeeves Algorithm

The Hooke-Jeeves algorithm is a member of the family of pattern search methods [146]. This algorithm comprises a combination of exploratory moves and pattern moves. An exploratory move aims to find the best point around the current point. In this move, first a base point, $\mathbf{x}_b^k = (x_1^k, x_2^k, ..x_n^k)$, is selected, and then each variable (x_i^k) is perturbed by a small amount $(x_i^k \pm \delta)$, and the objective function for a new point is evaluated. If the objective function is improved, the exploratory move is successful, and a new base point is reached. Otherwise, step length (δ) is reduced and the procedure is repeated. After exploratory moves, pattern moves are performed. In pattern moves, a new point is found using the current base point (best point found so far) and previous base point as follows:

$$\mathbf{x}_b^{k+2} = \mathbf{x}_b^k + (\mathbf{x}_b^{k+1} - \mathbf{x}_b^k) \qquad 4.15$$

where $\mathbf{x}_b^{k+2}$ is temporary base point for a new exploratory move. The Hooke-Jeeves algorithm is summarised as follows:

Hooke-Jeeves Algorithm

0. Select the initial base point, the increments (δ), a termination parameter $(\varepsilon > 0)$, and step reduction factor α > 1.
1. Perform exploratory moves for each variable. If the objective function's value is improved, go to 2. Otherwise, go to 3.
2. Perform pattern moves (Eq. 4.15). If the new point is found, set it as a new base point ($\mathbf{x}_b^{k+2}$). Go to 1 whatever the outcome is.
3. Check the stopping criterion $(\delta < \varepsilon)$. If not satisfied, set $\delta = {}^{\delta}/_{\alpha}$ and go to 1.

4.4.2.3 Nelder-Mead Algorithm

The last benchmark algorithm is the Nelder-Mead (NM) algorithm [147], which is a popular direct search method and can be applied for nonlinear optimisation problems. In a problem with n variables, this algorithm generates $n + 1$ vertices ($\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n, \mathbf{x}_{n+1}$.) to construct a simplex (i.e. a triangle with two variables) and then moves or reshapes this simplex to find the better solutions. To generate new vertices in a minimisation problem, the NM algorithm calculates the value of objective function associated with each vertex and replaces the vertex with the highest value of objective function (worst vertex) with a new vertex through a number of operations and using the centroid of the current simplex.

The algorithm includes three main operations: reflection, contraction of the simplex and expansion of the simplex. New vertices are generally constructed by reflecting the worst vertex to a new vertex. Additional mechanisms including expansion of the simplex and contraction of the simplex may be performed depending on the function's value of reflected vertex. The termination criterion is to check whether any orthogonal step leads to a further improvement of the objective function. As this algorithm may fail to converge, starting from different initial points could improve its efficiency [48]. The Nelder-Mead algorithm is summarised as follows:

Nelder- Mead

1. Compute the corresponding objective function for each point and sort points from the best ($\mathbf{x_1}$) to the worst ($\mathbf{x}_{\mathrm{n+1}}$) as follows:

$$f(\mathbf{x}_1) \leq f(\mathbf{x}_2) \ldots \leq f(\mathbf{x}_{\mathrm{n}}) \leq f(\mathbf{x}_{\mathrm{n+1}})$$

2. Calculate the centroid of all points except the worst point ($\mathbf{x}_{\mathrm{n+1}}$):

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{\mathrm{i=1}}^{\mathrm{n}} \mathbf{x}_{\mathrm{i}}$$

3. Compute the reflection of the worst point ($\mathbf{x}_{\mathrm{n+1}}$) as follows:

$$\mathbf{x}_{\mathrm{r}} = \ \bar{\mathbf{x}} \ + \ \alpha(\bar{\mathbf{x}} \ - \ \mathbf{x}_{\mathrm{n+1}})$$

where α is the reflection coefficient. Then, evaluate $f(\mathbf{x}_{\mathrm{r}})$. If $f(\mathbf{x}_1) \leq f(\mathbf{x}_{\mathrm{r}}) < f(\mathbf{x}_{\mathrm{n}})$, replace $\mathbf{x}_{\mathrm{n+1}}$ with $\mathbf{x}_{\mathrm{r}}$. Then go to 1.

4. If $f(\mathbf{x}_{\mathrm{r}}) < f(\mathbf{x}_1)$, compute the expansion point $\mathbf{x}_{\boldsymbol{e}}$ as follows:

$$\mathbf{x}_{\mathrm{e}} = \bar{\mathbf{x}} + \gamma(\mathbf{x}_{\mathrm{r}} - \bar{\mathbf{x}})$$

where γ is the expansion coefficient. Then, evaluate $f(\mathbf{x}_{\mathrm{e}})$. If $f(\mathbf{x}_{\mathrm{e}}) < f(\mathbf{x}_{\mathrm{r}})$, replace $\mathbf{x}_{\mathrm{n+1}}$ with $\mathbf{x}_{\mathrm{e}}$, otherwise replace $\mathbf{x}_{\mathrm{n+1}}$ with $\mathbf{x}_{\mathrm{r}}$. Then go to 1.

5. If $f(\mathbf{x}_{\mathrm{n}}) < f(\mathbf{x}_{\mathrm{r}}) < f(\mathbf{x}_{\mathrm{n+1}})$, compute the contraction point

$$\mathbf{x}_{\mathrm{c}} = \bar{\mathbf{x}} + \beta(\mathbf{x}_{\mathrm{r}} - \bar{\mathbf{x}})$$

where β is the contraction coefficient. Then, evaluate $f(\mathbf{x}_{\mathrm{c}})$. If $f(\mathbf{x}_{\mathrm{c}}) \leq f(\mathbf{x}_{\mathrm{n+1}})$, replace $\mathbf{x}_{\mathrm{n+1}}$ with $\mathbf{x}_{\mathrm{c}}$ and go to 1. Otherwise go to 6.

6. If $f(\mathbf{x}_{\mathrm{r}}) \geq f(\mathbf{x}_{\mathrm{n+1}})$, compute the contraction point $\mathbf{x}_{\mathrm{c}}$

$$\mathbf{x}_{\mathrm{c}} = \bar{\mathbf{x}} + \beta(\mathbf{x}_{\mathrm{n+1}} - \bar{\mathbf{x}})$$

Evaluate $f(\mathbf{x}_{\mathrm{c}})$. If $f(\mathbf{x}_{\mathrm{c}}) \leq f(\mathbf{x}_{\mathrm{n+1}})$, replace $\mathbf{x}_{\mathrm{n+1}}$ with $\mathbf{x}_{\mathrm{c}}$, and go to 1. Otherwise go to 7.

7. $\mathbf{x}_{\mathrm{i}} = \frac{\mathbf{x}_1 + \mathbf{x}_{\mathrm{i}}}{\mathbf{2}}$, for $2 \leq i \leq n + 1$. Then go to 8.

8. If the convergence criteria are not met, return to 1.

4.5 Results

The simulation-based optimisation methods were applied to building Type A in four diverse climates: Darwin, Brisbane, Melbourne and Hobart. Two different scenarios of this building are used for comparison of results. Scenario A is Building Type A as specified. The second scenario, Scenario B, is identical to Scenario A, but adds some energy efficiency measures. ABCB recommended building Type A as the representative of large commercial buildings in Australia [134, 135]. However, the original ABCB document was published in 2002. Therefore, it is expected that some energy efficiency measures have been applied to existing buildings over time. ABCB has also introduced many effective retrofit strategies to improve energy efficiency [148]. Some of these measures have been applied to simulation of this building as well such upgrading windows to the higher efficiency windows [128, 131, 149], shading installation [149], and lighting control system [128]. Therefore, four energy efficiency measures were considered in scenario B: 1) additional (0.5 meter) overhangs above windows; 2) double-pane windows ($\mathrm{U} = 2.678 \frac{\mathrm{W}}{\mathrm{m}^2}\mathrm{K}$, Solar Heat Gain Coefficient (SHGC) = 0.427 and Visible Transmittance (VT)$= 0.308$) instead of single-pane windows; 3) using daylighting control for each perimeter zone with one reference point with 320 lux set point at a height of 0.8 (m) from the floor and continuous lighting control (minimum electric power and light output $= 0$); and 4) removing temperature set back.

The objective function is to minimise the annual energy consumption of the building (Eq. 4.2) with respect to 15 variables listed in Table 4.1. The number of variables was selected as in [15], and the type and feasible search intervals were determined according to other similar studies [15, 17, 24, 40, 43-45]. It is worth noting that the inclusion of (likely) non-influential variables (e.g. roof emissivity) is informative for benchmarking optimisation algorithms as it shows how the algorithm handles them.

To conduct the building optimisation with benchmark algorithms, GenOpt optimisation software was used to perform optimisation with NM, PSOIW and PSO-HJ algorithms [150]. Ten optimisation runs of each method were conducted for each city. A High Performance Computing (HPC) cluster was used since between 3000 and 4600 building simulations were required for each run. The time required for each run with EnergyPlus 8.1.0 was between three and five days.

In order to provide a fair comparison among the different optimisation algorithms, the number of function evaluations (simulations) to achieve the optimised result was compared. In the hybrid PSO-HJ algorithm, PSO stops after the pre-defined number of iterations (3000 building simulations). However, Hooke-Jeeves terminates when no improvement is found (not after a set number of iterations). Thus, the number of simulations for each run was set in the following way. At first, the PSO-HJ algorithm was run to completion and the number of function evaluations was calculated. This number was considered as the stopping criterion for ACORs, NM and PSOIW (though the exact number of function evaluations will vary slightly due to the specifics of each algorithm).

An important factor in optimisation algorithm performance is the values for the free parameters. The parameters used in NM are those recommended in [24] and are shown in Table 4.2. The parameters used in the PSOIW and PSO-HJ algorithms are shown in Table 4.3. These parameters were set based on recommendations from previous studies that analysed PSO performance on benchmark functions and BOPs [44, 151]. The values for inertia weight in PSOIW and the values of parameters in HJ algorithm selected here were recommended by [24]. Parameters used in the ACOR recommended in [137] and are also shown inTable 4.4.

Table 4-1: Optimisation variables and their ranges

Variables	Description	Variable Range
X_1	Roof emissivity	[0.5-0.9]
X_2	Roof solar absorptance	[0.3-0.85]
X_3	Wall insulation (cm)	[1-10]
X_4	Wall solar absorptance	[0.3-0.9]
X_5	East window height (m)	[0.5-1.5]
X_6	North window height (m)	[0.5-1.5]
X_7	South window height (m)	[0.5-1.5]
X_8	West window height (m)	[0.5-1.5]
X_9	East overhang depth (m)	[0-1]
X_{10}	North overhang depth (m)	[0-1]
X_{11}	South overhang depth (m)	[0-1]
X_{12}	West overhang depth (m)	[0-1]
X_{13}	Heating setpoint (°C)	[18-22]
X_{14}	Cooling setpoint (°C)	[23-27]
X_{15}	Building orientation (degree)	[0-45]

Table 4-2: Parameters used for NM

NM parameters	Value
Accuracy	0.01
Step size factor	0.1
Block restart check	10
Modify stopping criterion	TRUE

Table 4-3: Parameters used for PSOIW and PSO-HJ

Parameters	**PSOIW**	**PSO-HJ**
Topology	Von Neumann	Von Neumann
Number of particles	100	100
Cognitive acceleration	2.05	2.05
Social acceleration	2.05	2.05
Constriction gain	-	1
Max velocity gain	0.2	0.2
Initial inertia weight	1.0	-
Final inertia weight	0	-
Mesh size divider	-	2
Initial mesh size exponent	-	0
Mesh size exponent increment	-	1
Number of step reductions	-	4

Table 4-4: Parameters used for ACOR (1) and ACOR (2)

Parameters	**ACOR (1)**	**ACOR (2)**
No. of new solutions used in each iteration (ants)	5	5
q parameter	0.0001	0.1
Speed of convergence (ξ)	0.85	0.85
Archive size	50	50

The optimisation results are presented in Table 4.5. The normalised energy consumption per unit floor area is presented to provide an easier comparison of results. Table 4.5 shows the best parameter sets among all ten runs for each algorithm in each city. For Brisbane, Hobart and Melbourne, the best solutions were obtained by ACOR

(1) after 3468, 4171 and 3372 building simulations, respectively. ACOR (2) found the best solution for Darwin after 3519 building simulations. In contrast, PSOIW found the worst solution for Hobart and Darwin after 3800 and 3600 building simulations, respectively. Likewise, NM found the worst solutions for Brisbane and Melbourne, respectively.

Table 4.5 also shows that the optimised building orientations are approximately zero degrees for Darwin, Hobart and Melbourne and almost ten degrees relative to North (clockwise) for Brisbane. For all cities, the optimum wall has the minimum solar absorptance, and best roof has the maximum emissivity with minimum solar absorptance. The optimised wall insulation thickness is 1 cm ($U_{wall} = 1.88\ \mathrm{W/m^2}$K). The algorithm's selection of the minimum allowable insulation thickness can be explained as follows. The HVAC system operates only during the daytime and the internal loads are fairly high. Due to this combination of usage factors and the relatively mild Australian climates, the dominant mode of operation of the HVAC system is cooling, even in winter. Therefore, increasing the insulation thickness will lead to higher cooling loads in winter months, which more than offsets any reductions in the cooling load in the summer months [152]. For example, if the optimised insulation thickness increases 1 cm (10% of the allowable range), the annual cooling loads increase $33\ (\mathrm{GJ})$, $11.35\ (\mathrm{GJ})$, $21\ (\mathrm{GJ})$ and $16\ (\mathrm{GJ})$ for Brisbane, Darwin, Hobart and Melbourne, respectively, while the heating loads decrease only 3.3 (GJ) for both Hobart and Melbourne. The optimum window and overhang values depend on city and building direction because of the trade-off between lighting, cooling and heating loads. These results can also be used to compute the optimised values for window-to-wall ratio. For example, Melbourne has window-to-wall ratios (excluding plenum) of 27.7%, 32.7%, 37.2% and 31.8% for the East, North, South and West building faces, respectively. The minimum and maximum were selected for heating and cooling set-points for all cities, respectively. This is clearly expected when thermal comfort is not considered in the objective function and only as a constraint on the allowable range of indoor temperature set points. It should be noted that Table 4.5 shows optimisation solutions with decimal points, which are important in terms of solutions quality of optimisation algorithms, but this might be impractical for some variables in building design. For example, the heating/cooling set points are likely be rounded to their nearest integer in building design.

From an energy point-of-view, the difference between optimised objective functions obtained by ACOR (e.g. $642.56\ \mathrm{MJ/m^2}$ (Brisbane)) and PSO-HJ (e.g. $642.74\ \mathrm{MJ/m^2}$ (Brisbane)) are small. As can be seen in this table, despite slight differences between optimised objective functions, significantly different sets of parameters have been obtained by each algorithm, showing that the building objective function is very multi-modal. This fact provides building designers with more options in designing low energy buildings.

In real world optimisation problems, it is very likely that few optimisation runs will be utilised due to the high computational cost. Therefore, an algorithm that consistently leads to good solutions is preferable. A low mean value with small variability in results suggests a more reliable algorithm. Box–Whisker (BW) plots display the distribution of optimisation results of annual energy consumption ($\mathrm{MJ/m^2}$) for each city, based on ten runs. Comparing the median values in figures 4.3- 4.6 shows that ACOR (2) and ACOR (1) perform the best for all cites, respectively. Although the median value of ACOR (1) is very close to ACOR (2), it has a larger variability than the ACOR (2), which makes ACOR (1) less reliable than ACOR (2). In contrast to ACOR, in all cities the spread of the optimisation results in NM is much larger than others. In addition, the median values of NM are also greater than other algorithms except for Darwin where PSOIW is highest. Apart from NM, the spread of the optimisation results in PSO-HJ for Brisbane and Hobart is larger than others.

Table 4-5: Optimisation results, best solution of each algorithm

	Algorithm	Objective Function (MJ/m^2)	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15
Brisbane	**NM**	644.21	0.66	0.49	1	0.30	0.75	0.75	0.86	0.74	0.57	0.94	0.54	0.88	21.58	27.00	19.65
	PSOIW	644.17	0.69	0.33	1	0.30	0.78	0.87	1.04	0.68	0.71	0.62	0.63	0.95	20.08	26.98	4.77
	PSO-HJ	642.74	0.90	0.30	1	0.30	0.75	0.79	0.93	0.75	1.00	0.60	0.58	1.00	18.00	27.00	2.60
	ACOR (1)	**642.56**	**0.90**	**0.30**	**1**	**0.30**	**0.75**	**0.75**	**0.95**	**0.75**	**1.00**	**0.65**	**0.72**	**1.00**	**18.43**	**26.99**	**10.00**
	ACOR (2)	642.74	0.90	0.30	1	0.30	0.86	0.83	0.94	0.75	0.74	0.71	0.71	1.00	19.04	26.99	11.66
Darwin	**NM**	780.11	0.69	0.30	1	0.30	0.75	0.70	0.88	0.74	1.00	0.78	0.75	0.97	18.64	27.00	15.29
	PSOIW	781.32	0.84	0.31	1	0.30	0.69	0.67	0.93	0.74	1.00	0.89	0.71	0.92	21.31	26.98	36.72
	PSO-HJ	780.11	0.90	0.30	1	0.30	0.75	1.00	0.75	0.75	1.00	0.79	0.57	1.00	20.50	27.00	13.44
	ACOR (1)	779.25	0.90	0.30	1	0.30	0.73	0.75	0.91	0.75	1.00	1.00	0.69	1.00	21.92	26.99	2.01
	ACOR (2)	**779.24**	**0.90**	**0.30**	**1**	**0.30**	**0.72**	**0.75**	**0.90**	**0.75**	**1.00**	**1.00**	**0.68**	**1.00**	**21.18**	**26.99**	**0.02**
Hobart	**NM**	547.10	0.90	0.39	1	0.30	1.00	0.67	1.36	0.88	0.76	0.53	0.60	0.76	18.02	27.00	18.28
	PSOIW	547.10	0.74	0.48	1	0.30	1.11	0.92	1.16	0.93	0.81	0.76	0.47	0.77	18. 00	27.00	17.95
	PSO-HJ	546.13	0.90	0.30	1	0.30	0.95	1.07	1.34	1.02	0.78	0.80	0.52	0.77	18.00	27.00	7.25
	ACOR (1)	**545.92**	**0.90**	**0.30**	**1**	**0.30**	**0.75**	**1.02**	**1.26**	**0.92**	**1.00**	**0.77**	**0.25**	**0.70**	**18.00**	**27.00**	**0.00**
	ACOR (2)	545.95	0.90	0.30	1	0.30	0.89	1.05	1.27	0.92	0.77	0.80	0.28	0.77	18.00	27.00	8.85
Melbourne	**NM**	577.19	0.67	0.62	1	0.30	0.74	0.89	0.96	0.79	0.56	0.56	0.59	0.69	18.57	26.99	4.84
	PSOIW	576.44	0.83	0.37	1	0.30	0.86	0.80	1.03	0.82	0.70	0.65	0.45	0.71	18.20	27.00	13.27
	PSO-HJ	575.82	0.90	0.30	1	0.30	0.78	0.75	0.99	0.75	0.68	1.00	0.28	1.00	18.50	27.00	9.60
	ACOR (1)	**575.58**	**0.89**	**0.30**	**1**	**0.30**	**0.75**	**0.88**	**1.01**	**0.86**	**1.00**	**0.76**	**0.39**	**0.74**	**18.30**	**27.00**	**0.00**
	ACOR (2)	575.64	0.90	0.30	1	0.30	0.88	0.87	0.99	0.75	0.76	0.75	0.38	1.00	18.30	27.00	19.70

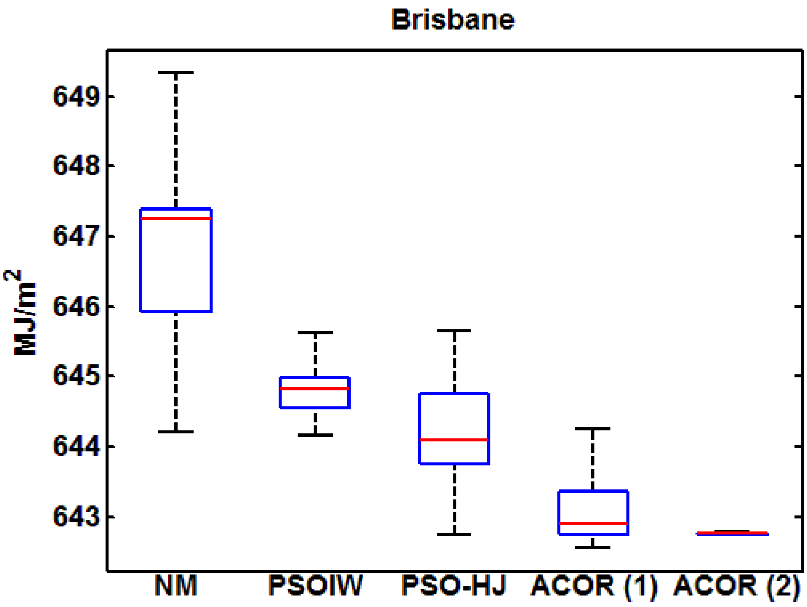

Figure 4.3: Algorithm comparison results for Brisbane

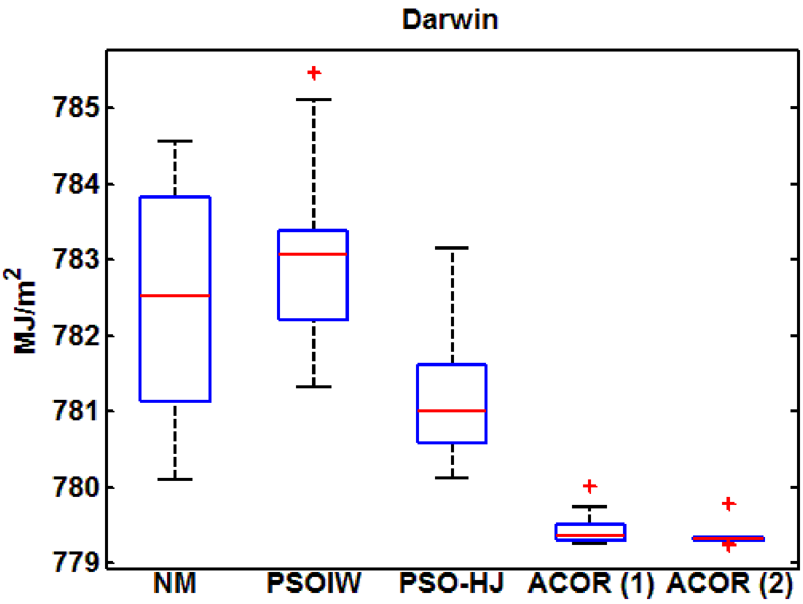

Figure 4.4: Algorithm comparison results for Darwin

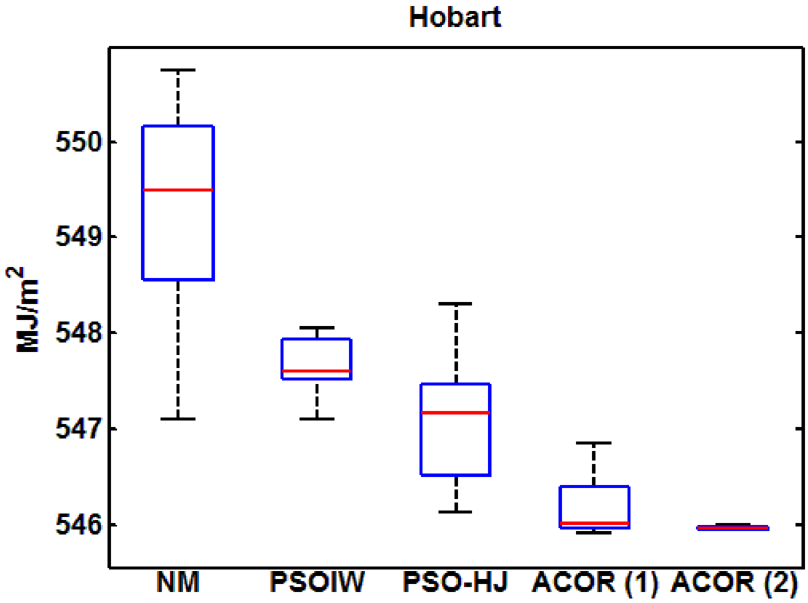

Figure 4.5: Algorithm comparison results for Hobart

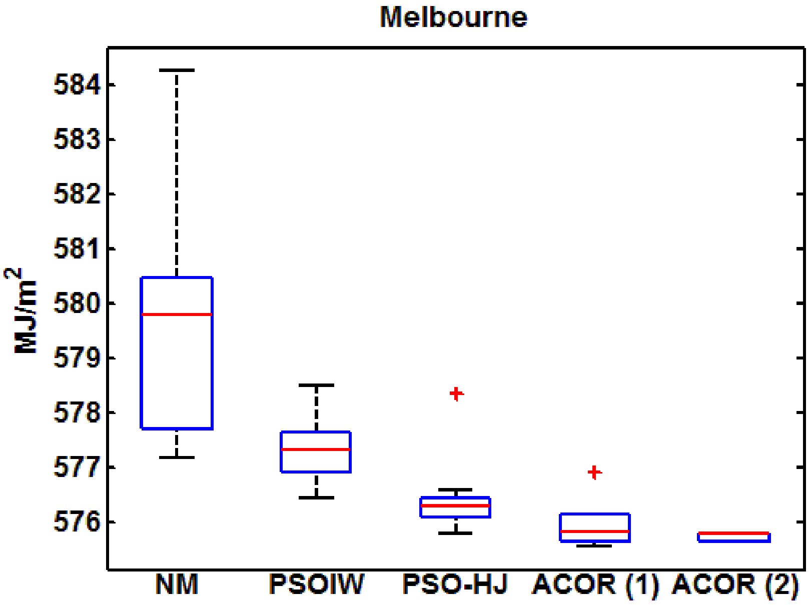

Figure 4.6: Algorithm comparison results for Melbourne

The Wilcoxon rank-sum test was applied to understand the statistical significance of the differences in the algorithms' performance. The Wilcoxon rank-sum is a non-parametric statistical hypothesis test used to understand the probability that the difference between two groups (here two algorithms) is significant. In this test, low p-values indicate a low probability that the results were obtained by random chance while high p-values indicate a significant probability that there is no difference between the algorithm performances. Table 4.6 shows that for all cities the differences between both ACOR algorithms and NM, PSO-HJ as well as PSOIW, are very significant. There is, however, no significant difference between ACOR (2) and ACOR (1).

Table 4-6: Wilcoxon rank-sum test results. Bold numbers indicate p-values that are below the conventional 0.05 significance level

Algorithms	**Brisbane (P-value)**	**Darwin (P-value)**	**Hobart (P-value)**	**Melbourne (P-value)**
ACOR (2) VS NM	(**0.0001**)	(**0.0001**)	(**0.0001**)	(**0.0001**)
ACOR (2) VS PSOIW	(**0.0001**)	(**0.0001**)	(**0.0001**)	(**0.0001**)
ACOR (2) VS PSO-HJ	(**0.0022**)	(**0.0001**)	(**0.0001**)	(**0.0003**)
ACOR (2) VS ACOR (1)	(0.2730)	(0.1405)	(0.1405)	(0.4274)
ACOR (1) VS NM	(**0.0002**)	(**0.0001**)	(**0.0001**)	(**0.0001**)
ACOR (1) VS PSOIW	(**0.0002**)	(**0.0001**)	(**0.0001**)	(**0.0003**)
ACOR (1) VS PSO-HJ	(**0.0173**)	(**0.0001**)	(**0.0022**)	(**0.0173**)

Another important metric for optimisation algorithms is the convergence rate. In building optimisation problems, the evaluation of objective function is time-consuming, and it is therefore crucial that the number of function evaluations is kept to a minimum. Comparing convergence speed of optimisation algorithms is particularly important when the overall performance is very close in terms of the objective value (as is the case here).

Figure 4.7 shows an example of the optimisation run (for a solution close to the median) for Brisbane. As can be seen, both ACOR (1) and ACOR (2) converge to their final solutions much faster than other metaheuristic algorithms. In early iterations, NM performance is better than PSOIW and PSO-HJ and quickly converges to a solution. However, its final solution is quite far from the best found solution. It can also be seen in the hybrid PSO-HJ algorithm, the PSO stopped after 3000 building simulations and then HJ refined the PSO results. The overall convergence speed of optimisation algorithms after ten runs is shown in figures 4.8- 4.11.

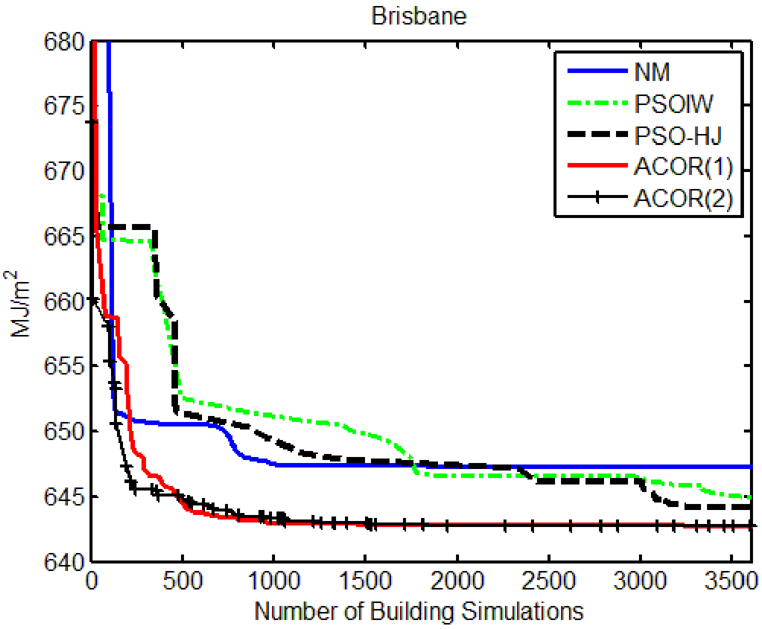

Figure 4.7: Convergence speed for the solution close to median in Brisbane

Figure 4.8 and Figure 4.9 compare the convergence speed in the final stages of optimisation when algorithms converge to a solution very close to the final (e.g. within 0.1%) for Brisbane and Darwin. As can be seen, NM produced highly inconsistent results. In the PSO-HJ results, the solutions were found when the HJ algorithm started refining PSO solutions (after 3000 iterations). A comparison of median values shows that both ACOR (1) and ACOR (2) are between two to four-and-a-half times faster than NM, PSOIW and PSO-HJ. Figure 4.10 and Figure 4.11 compare the convergence speed in the initial optimisation stages when algorithms converge to a solution close to the optimised (e.g. within in 1%) for Hobart and Melbourne. Both ACOR algorithms showed slightly faster convergence rates than NM and much faster performance than PSOIW and PSO-HJ. A comparison of median values shows that ACOR (1) is almost seven times faster than PSO-HJ in Melbourne, and although NM has a potentially fast convergence rate, this rate is inconsistent and the solutions found have significantly higher energy consumption than the ACOR solutions.

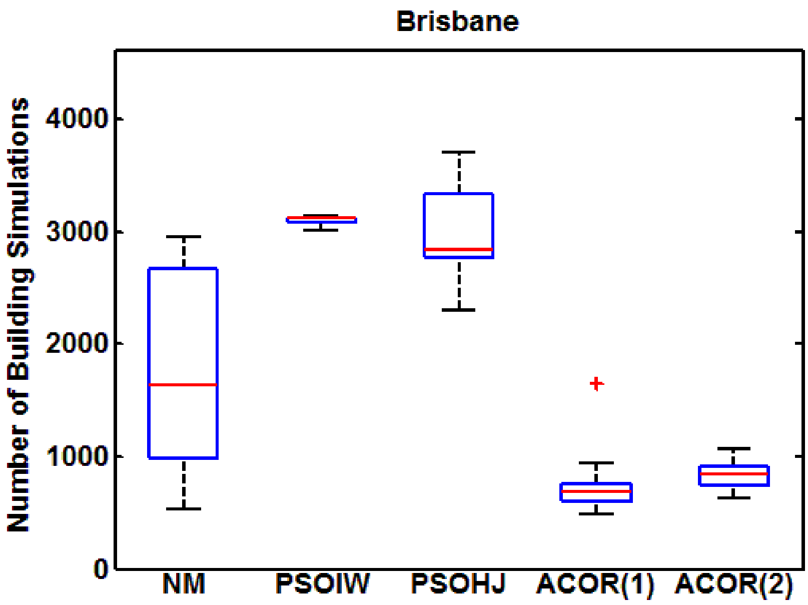

Figure 4.8: Number of building simulations required for each algorithm to converge to within 0.1% of the final solution for Brisbane

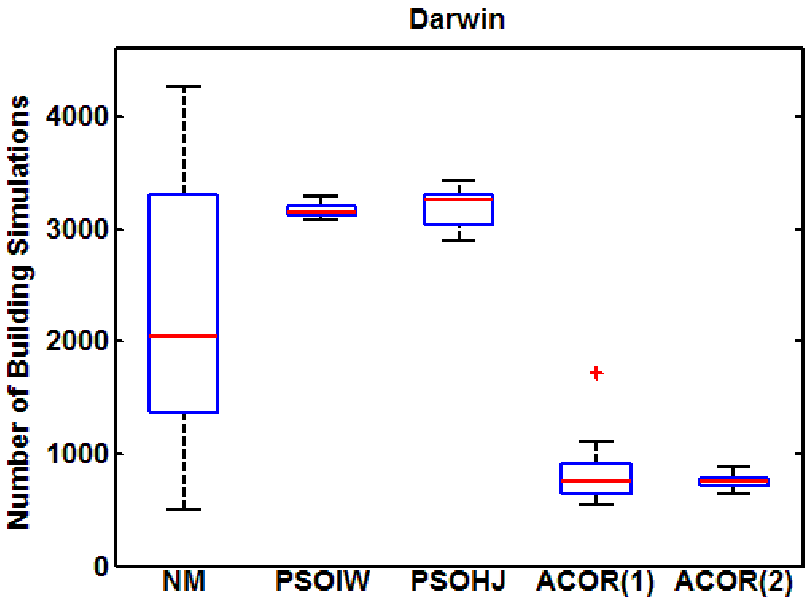

Figure 4.9: Number of building simulations required for each algorithm to converge to within 0.1% of the final solution for Darwin

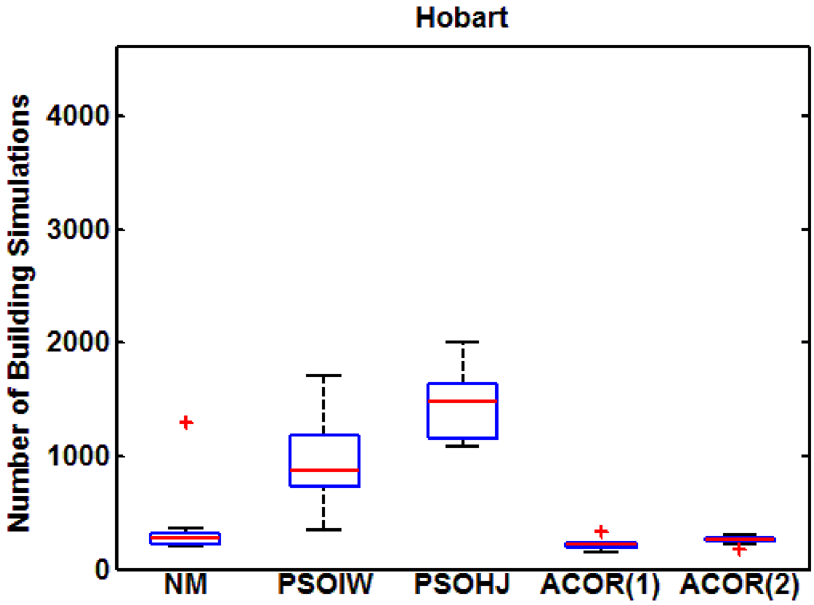

Figure 4.10: Number of building simulations required for each algorithm to converge to within 1% of the final solution for Hobart

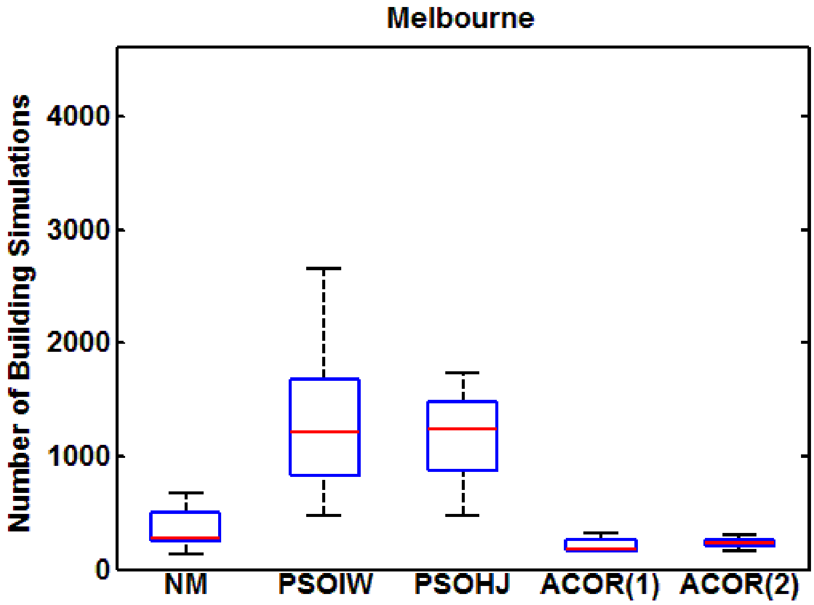

Figure7b

Figure 4.11: Number of building simulations required for each algorithm to converge to within 1% of the final solution for Melbourne

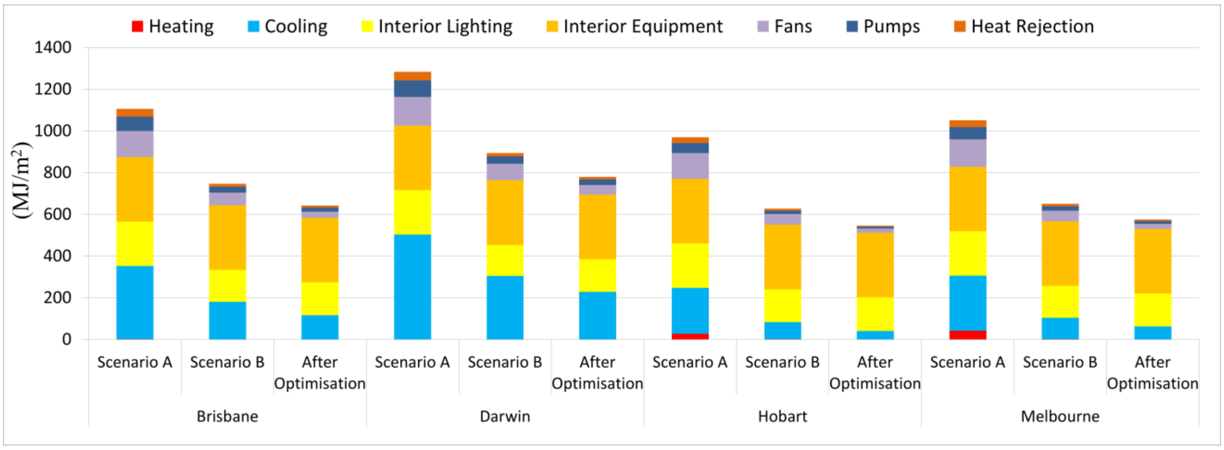

Figure 4.12: Building annual energy consumption for Scenarios A, B, and after optimisation

Figure 4.12 shows the building annual energy consumption and the breakdown of energy consumption for Scenarios A, B, and after optimisation. This figure also shows that cooling loads in Scenario B in comparison to Scenario A reduced by 48.4%, 39.5%, 62.6% and 61.1% for Brisbane, Darwin, Hobart and Melbourne, respectively. After applying simulation-based optimisation, the annual energy consumption (compared to Scenario B) was reduced by 13.9%, 12.9%, 12.9% and 11.47% for Brisbane, Darwin, Hobart and Melbourne, respectively. Comparison of the energy breakdown between Scenario B and optimised building shows that optimisation has significantly reduced the fan and cooling loads (fan energy consumption fell 53.45%, 43.37%, 61.32% and 53.22% for Brisbane, Darwin, Hobart and Melbourne, respectively). The optimised building design in Darwin saw the maximum fan energy reduction by 34.65 $\mathrm{MJ/m^2}$. More importantly, cooling loads were reduced by 35.7%, 24.9%, 52.03% and 39.5% for Brisbane, Darwin, Hobart and Melbourne, respectively. Darwin and Hobart experienced the maximum and minimum cooling load reductions of 75.92 $\mathrm{MJ/m^2}$ and 42.79 $\mathrm{MJ/m^2}$, respectively. It should be noted that despite the use of dimming electric lighting to harvest daylighting, lighting loads almost remain constant between Scenario B and the optimised result. Since minimising of the cooling and lighting loads are conflicting objectives, it is noteworthy that the optimisation algorithm prioritises reduction of the cooling loads, which is not surprising in Australia (where cooling loads are typically high). Since the optimisation seeks the best balance between the various building loads, it is highly likely that an attempt to further decrease the lighting or cooling load would lead to a corresponding increase of equal or greater magnitude in the others.

4.6 Conclusion

In this chapter, an ACOR algorithm was developed for solving building optimisation problems and was applied to optimise fifteen variables in a representative commercial building in four diverse climates in Australia. A comparison between ACOR and three benchmark algorithms, NM, PSOIW and PSO-HJ, established the supremacy of ACOR in solving BOPs. All algorithms found good solutions. However, the two different parameter settings for ACOR (ACOR (1) and ACOR (2)) found results that are closer to global optimum than PSOIW and PSO-HJ. In terms of consistency (spread of results), ACOR (2) showed less variation in results and was by far more consistent than other algorithms. Importantly, both ACOR (1) and ACOR (2) converged much faster to their final solutions than the PSOIW and PSO-HJ. Indeed, since computational cost is a key issue limiting BOP practicality, this represents a significant result. The Wilcoxon rank-sum test confirmed that the superior performance of ACOR over the two other algorithms was statistically significant. Overall, ACOR (2) is recommended for solving BOPs due to finding more precise solutions, greater consistency in results and a fast convergence rate.

This chapter also highlights the importance of using simulation-based optimisation for commercial buildings in Australia. The results showed that building optimisation using a limited set of variables can achieve energy reductions of at least 11.47% and up to 13.9%, even after implementing the energy saving measures of Scenario B. This reduction was achieved largely by reducing the cooling load without significantly altering the lighting requirements (see Figure 4.12). Applying a simulation-based optimisation on an Australian representative ten-storey commercial building identifies the potential energy saving solutions, provides a better understanding of optimised values of design variables, and helps building designers meet building code requirements to design low-energy buildings in Australia.

Chapter 5: Algorithm for Mixed Variables

5.1 Overview

Many building optimisation problems include a combination of continuous and categorical (discrete) decisions, such as optimising the overhang size (e.g. continuous variable) and the window glazing type (e.g. categorical variable) [153, 154]. In this chapter, the ant colony optimisation algorithm with capabilities for handling mixed variables (ACOMV) is applied to BOPs for the first time. Motivated by ACOMV results, the modified version of ACOMV called ACOMV-M is developed for the first time. The results of ACOMV-M algorithms are then presented and compared against the benchmark algorithm identified in the literature.

This chapter is organised as follows: Section 5.2 details the problem statement including both continuous and categorical variables. Section 5.3 details the ACOMV-M algorithm. Section 5.4 presents the case study and Section 5.5 presents the results, followed by a chapter conclusion in Section 5.6.

5.2 PROBLEM STATEMENT

The building optimisation problem considered here can be formally stated as

$$\begin{gathered}\min_{\mathbf{x}} \ f(\mathbf{x}) \\ \text{subject to: } \mathbf{x} \in \mathbb{X} \ \subseteq \mathbb{R}^r \times \mathbb{V}^c\end{gathered} \qquad 5.1$$

where $f(\cdot)$ is the objective function, and $\mathbb{X}$ is the feasible space of independent design variables, composed of continuous and categorical subspaces ($\mathbb{R}^r$ and $\mathbb{V}^c$, respectively). For each of the continuous variables (R^i) in $\mathbf{x}$ the feasible design space is simply stated in terms of upper and lower bounds on parameters: $-\infty < \ell^i \leq \mathrm{R}^i \leq u^i < +\infty,\ i = 1,2, \dots, r$ where ℓ^i and u^i are the lower bound and the upper bound of the ith optimisation variable. Since the decision variable input ranges can be normalised, we may assume that $\ell = 0$ and $u = 1$. For each of the categorical variables in $\mathbf{x}$, the feasible design space is limited to a finite set of t_i values $\mathbb{V} \in \{v_1, \ldots, v_{t_i}\}$. The objective function $f(\cdot)$ is the building annual end use energy consumption ($\mathrm{MJ/m^2}$ Year), which can be written as follows:

$$f(\mathbf{x}) = E_c(\mathbf{x}) + E_f(\mathbf{x}) + E_\ell(\mathbf{x}) + E_p(\mathbf{x}) + E_h(\mathbf{x}) + E_m(\mathbf{x}) \tag{5.2}$$

where E_c is the energy consumption for space cooling, E_f is the energy consumption of the fans, E_ℓ is the energy consumption of lighting, E_p is the energy consumption of pumps, E_h is the energy consumption for space heating and E_m is the energy consumption that includes interior equipment and heat rejection.

5.3 Ant Colony Optimisation for Mixed Variables

Ant Colony Optimisation for Mixed Variables (ACOMV), developed in [155], is an extended version of ACOR for solving optimisation problems with both continuous and categorical (discrete) variables.

R_1^1	R_1^2	...	R_1^r	C_1^1	...	C_1^c
R_2^1	R_2^2	...	R_2^r	C_2^1	...	C_2^c
$\vdots$	$\vdots$	...	$\vdots$	$\vdots$	...	$\vdots$
R_j^1	R_j^2	...	R_j^r	C_{j}^1	...	C_{j}^c
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
R_M^1	R_M^2	...	R_M^r	C_{M}^1	...	C_{M}^c

$f(\mathbf{x}_1)$	ω_1
$f(\mathbf{x}_2)$	ω_2
$\vdots$	$\vdots$
$f(\mathbf{x}_j)$	ω_j
$\vdots$	$\vdots$
$f(\mathbf{x}_M)$	ω_M

Figure 5.1: Solution archive for ACOMV (adapted from [155])

ACOMV, similar to ACOR, operates on a *solution archive*, an example of which is shown in Figure 5.1. This archive is generated randomly and contains two groups of columns, one for each of the r categorical variables, and another for the c continuous variables. A solution j is thus represented as a $r + c$ dimensional vector, $\mathbf{x}_j = [R_j^1, R_j^2, \dots, R_j^r, C_j^1, C_j^2, \dots, C_j^c]$ with an associated objective function value $f(\mathbf{x}_j)$. Solutions in the archive are sorted from lowest to highest objective values, i.e.

$$f(\mathbf{x}_1) \leq f(\mathbf{x}_2) \leq \dots \leq f(\mathbf{x}_j) \leq \dots \leq f(\mathbf{x}_M) \tag{5.3}$$

For the continuous variables in $\mathbf{x}$, new candidate solutions are generated according to a Gaussian kernel probability density function (PDF) based on the solutions in the archive

$$G^i(x) = \sum_{j=1}^{M} \omega_j \, g_j^i(x) = \sum_{j=1}^{M} \omega_j \frac{1}{\sigma_j^i \sqrt{2\pi}} \, e^{-\frac{\left(x-\mu_j^i\right)^2}{2\sigma_j^{i2}}} \tag{5.4}$$

where $G^i(x)$ is the Gaussian kernel for the ith dimension of the solution. For the continuous variable i of solution j, $g_j^i(x)$ is the sub-Gaussian function, while μ_j^i and σ_j^i are mean value and the standard deviation, respectively. The *weights* ω_j are set so that solutions with lower objective values are preferred, since they likely indicate neighbourhoods where good solutions may be found. Like ACOR, the weights are assigned based on the position of a solution in the archive

$$\omega_j = \frac{1}{qM\sqrt{2\pi}} \, e^{-\frac{(j-1)^2}{2q^2M^2}} \tag{5.5}$$

where q is a free parameter that controls how sharply the weights decrease with the archive index j. Low values of q increase the weights of the best solutions relative to the other solutions in the archive.

The mean and standard deviation of the of the sub-Gaussians are also set based on the archive solutions

$$\mu_j^i = R_j^i \qquad 5.6$$

$$\sigma_j^i = \xi \sum_{\ell=1}^{M} \frac{\left|R_\ell^i - R_j^i\right|}{M-1} \qquad 5.7$$

In other words, the standard deviation is set according to the average distance of $\mathbf{x}_j$ from the other $M-1$ solutions in the archive along dimension i in the parameter space. The free parameter ξ is simply a scaling factor, which allows users to set the percentage of this average.

The new candidate solutions are generated according to the distribution in Eq. 5.4 via a two-stage process. First, a solution from the archive is randomly selected with probability

$$P_j = \frac{\omega_j}{\sum_{r=1}^{M} \omega_r} \qquad 5.8$$

Then, a new solution is sampled using the selected Gaussian function

$$g_j^i(\tilde{x}^i) = \frac{1}{\sigma_{\mathrm{j}}^i \sqrt{2\pi}} e^{-\frac{\left(\tilde{x}^i - \mu_{\mathrm{j}}^i\right)^2}{2\sigma_{\mathrm{j}}^{i2}}} \qquad i = 1,2,\ldots,N \qquad 5.9$$

where j is the selected solution from the archive and N is the number of elements in each solution ($N = r + c$). The objective value of this solution is then evaluated and the generation procedure repeats until m candidate solutions are generated. The archive is then updated by selecting the best M solutions from the $M + m$ solutions. Prior to optimisation with the ACOMV algorithm, all variables are normalised between zero and one ($l_i = 0$ and $u_i = 1$). However, a variable (R^i) may violate the domain boundary constraint during the generation of new solutions. If this occurs, R^i is repaired as follows:

$$\text{if } R^i < 0 \rightarrow \quad R^i = |R^i| \qquad 5.10$$

$$\text{if } R^i > 1 \rightarrow \quad R^i = 1 - (R^i - floor(R^i)\,)$$

For each categorical variable ($1 \leq i \leq c$), each solution is incrementally constructed by randomly choosing one of the t_i available values $v_\ell^i \in \{v_1^i, \ldots, v_{t_i}^i\}$. The probability of choosing the ℓth value is

$$p_\ell^i = \frac{\omega_\ell}{\sum_{j=1}^{t_i} \omega_j} \qquad 5.11$$

where ω_ℓ is the weight associated to the ℓth available value. The weight ω_ℓ is calculated as

$$\omega_\ell = \begin{cases} \frac{\omega_{j_\ell}}{u_\ell^i} + \frac{q}{\eta}, & if(\eta > 0,\ u_\ell^i > 0) \\ \frac{\omega_{j_\ell}}{u_\ell^i}, & if(\eta = 0,\ u_\ell^i > 0) \\ \frac{q}{\eta}, & if(\eta > 0,\ u_\ell^i = 0) \end{cases} \qquad 5.12$$

In the above equation ω_{j_ℓ} is calculated according to Eq. 5.5, and j_ℓ is the index of the highest quality solution that uses value v_ℓ^i for the categorical variable i. u_ℓ^i is the number of solutions that use value v_ℓ^i for the categorical variable i in the solution archive. The parameter η is the number of values from the t_i available ones that are not used by the solutions in the archive, and q is the same parameter of the algorithm that was used in Eq. 5.5. To avoid stagnation, ACOMV may use a restart strategy, meaning that if the number of consecutive iterations with no improvement is larger than a predefined number ($MaxStagIter$), the algorithm restarts, while keeping the best solution so far.

A key challenge in the application of any optimisation algorithm is striking the proper balance between exploration of the search space and intensification of the search near optimised solutions. In ACOMV, this behaviour is controlled using the parameters q and ξ. Smaller values of q promote intensification by assigning relatively large weights to better solutions in the archive and thus generating more candidate solutions in the neighbourhood of the best solutions. Larger values of q increase exploration by assigning more uniform weights to solutions in the archive. The parameter ξ is a normalised width of the sub-Gaussians, in which its higher values promote increased exploration around a given solution, while its lower values increase intensification near it.

Motivated by the above observations and an initial simulation study (detailed in the next section), a modified version of ACOMV was developed based on the observations of the effect of the parameter q. In the modified algorithm (ACOMV-M), q is decreased automatically when the number of consecutive iterations with no improvement exceeds a specific number ($Shrinkage\_Factor$). This mechanism (Eq. 5.13) helps the algorithm search locally to refine the solution near the best solutions. The ACOMV-M algorithm is summarised below.

$$\begin{cases} q = 0.05099, & if\ \ Shrinkage\ Factor < 2 \\ \quad q = 0.0001, & if\ \ Shrinkage\ Factor \geq 2 \end{cases} \qquad 5.13$$

ACOMV-M algorithm

```
1. Select values for the parameters $q$, $\xi$, $MaxStagIter$, $Shrinkage\_Factor$, $M, m \leq M$
2. Initialise. Randomly generate $\mathbf{x}_j$, $j = 1,2,\ldots,M$ according to component-wise uniform distributions between the upper and lower bounds. Store all solutions in the solution archive and compute the objective value associated with each solution
3. Sort solutions in ascending order according to their objective values so that Eq. 5.3 is satisfied
4. Calculate weights according to Eq. 5.5
5. Generate a new solution
    a. Select a solution $j$ from the archive with probabilities from Eq. 5.8
    b. Generate a continuous solution according to Eq. 5.9
    c. Adjust any variable values violating constraints according to Eq. 5.10
    d. Generate a categorical solution according to Eq. 5.11 and Eq. 5.12
6. Repeat  step 5, $m$ times
7. Evaluate objectives of $m$ new solutions
8. Select the best $M$ solutions from the $M + m$ solutions available
9. Check the restart condition ($MaxStagIter$). If it is satisfied, initialise archive while keeping the best-so-far solution
10. Check the shrinkage condition (Eq.5.13)
11. Check the stopping criterion. If it is not satisfied, return to 3.
```

5.4 Case Study

The simulation-based optimisation methods were applied to building Type B. This building is also recommended by ABCB as the typical medium-size commercial buildings and has all features of a real building. This building has been used in many studies [102, 120, 131, 149]. Therefore, it is a suitable case study and it provides an opportunity to test the proposed optimisation methods in another case study as well. in another case study as well. In addition, building Type B is computationally more efficient since its simulation time is approximately thirty percent faster than a Type A building. Two diverse climates were considered here: Brisbane with warm humid summers and mild winters, and Hobart with mild to warm summers and cold winters [124]. The typical number of climates applied to algorithm comparison studies varies from one [43, 48] to three [24, 44]. Therefore, two different climates were considered here upon the condition that the benchmarking results are consistent. Details of building Type B were stated in Chapter 3. However, two modifications were made for this building before optimisation. First, daylighting control for each perimeter zone with one reference point with 320 lux set point at a height of 0.8 (m) from the floor with continuous lighting control (minimum electric power and light output $=\ 0$) was added. Secondly, temperature set back was removed.

The objective function was to minimise the annual energy consumption of the building (Eq.5.2). Optimised values of variables (presented in chapter 4) showed that for the different Australian climates maximum solar emissivity and minimum solar absorbtance for walls and roofs are needed even for Hobart (with mild to warm summers and cold winters). In addition, for the retrofit purposes, considering building orientation as an optimisation variable is not an appropriate assumption. Accordingly, in this chapter, nine optimisation variables which are listed in Table 5.1 and Table 5.2

were selected among fifteen optimisation variables presented in Table 5-1. Different sets of these nine variables (including both continuous and discrete variables) have been used in other studies as well [15, 24, 44, 45]. In the next section, in order to investigate the performance of the optimisation algorithms and identify the best one, the results of three algorithms (i.e. ACOMV, ACOMV-M and PSOHJ) are compared.

Table 5-1 : Optimisation variables and their ranges

Variables	Description	Variable Range
x_1	Wall insulation (cm)	[1- 10]
x_2	North overhang depth (m)	[0-1.2]
x_3	South overhang depth (m)	[0-1.2]
x_4	East overhang depth (m)	[0-1.2]
x_5	West overhang depth (m)	[0-1.2]
x_6	North window	Table 5.2
x_7	South window	Table 5.2
x_8	East window	Table 5.2
x_9	West window	Table 5.2

Table 5-2: Different window types used for categorical variables (X_6 to X_9)

Window	Type	U Value [W/m^{2}K]	SHGC	Visible Transmittance
Single glazed	1-Clear	5.88	0.81	0.88
	2-Tinted	5.77	0.60	0.43
	3-Reflective	5.06	0.40	0.30
	4-Low-e	3.43	0.63	0.84
Double glazed	5-Clear	2.71	0.70	0.78
	6-Reflective	2.46	0.30	0.27
	7-Tinted	2.69	0.48	0.38
	8-Low-e	1.77	0.57	0.74
	9-Low-e-tinted	1.77	0.38	0.44
Triple glazed	10-Clear	1.76	0.61	0.69
	11-Low-e	1.30	0.51	0.66

5.5 Results

The performance of the optimisation algorithms are compared with three key performance metrics: 1) quality of solutions, 2) consistency (reliably achieving solutions close to the optimised), and 3) computational cost. Twenty optimisation runs of each algorithm were conducted to provide a statistical characterisation of their performance. In order to provide a fair comparison among the different algorithms, the total number of building simulations was kept constant. A limit of 4000 building simulations (i.e. stopping criterion) was selected for ACOMV algorithms. However, the hybrid PSOHJ algorithm cannot be limited in the same way, since the Hooke-Jeeves algorithm stops only when no improvement is found. Therefore, a limit of 3800 building simulations was selected for PSO, and then Hooke-Jeeves refines the PSO results after 100 to 200 building simulations. A High Performance Computing (HPC) cluster was used, and the time required for each optimisation run with EnergyPlus $8.1.0$ is approximately 50 hours.

An important factor in optimisation algorithm performance is the values for the free parameters. Table 5.3 shows the parameters used in the PSOHJ algorithms. These parameters were set based on recommendations from previous studies, which analysed PSO performance on benchmark functions and building optimisation problems [44, 151]. Except for $MaxStagIter$, all parameters used in the ACOMV are those recommended in [155] and are used in this study and shown in Table 5.4. The parameter $MaxStagIter$ is set to 400 (approximately 10% of the maximum number of iterations) based on trial-and-error. In ACOMV-M, the $Shrinkage\_Factor$ is set to $2 \times MaxStagIter$.

Table 5-3: Parameters used for PSOHJ

Parameters	PSOHJ
Topology	Von Neumann
Number of particles	100
Cognitive acceleration	2.05
Social acceleration	2.05
Constriction gain	1
Max velocity gain	0.2
Mesh size divider	2
Initial mesh size exponent	0
Mesh size exponent increment	1
Number of step reductions	4

Table 5-4: Parameters used for ACOMV

Parameters	ACOMV
No. of new solutions used in each iteration (ants)	5
q parameter	0.0509
Speed of convergence (ξ)	0.6795
Archive size	90
Stagnating iterations before restart ($MaxStagIter$),	400

Table 5.5 shows the quality of solutions found by each algorithm for both Brisbane and Hobart. This table presents the best solutions, and solutions that are close to median value for each algorithm among twenty runs. To facilitate comparison among the results, the energy consumption per unit floor area has been presented. As can be

seen, the maximum energy reduction was obtained by ACOMV-M, however, PSOHJ found solutions very close to the best solution. For this building, the difference of solutions between worst and the best algorithm is less than 0.5%, although this difference will likely vary significantly for different buildings and/or climates.

In order to compare the consistency of optimisation algorithms, a Box–Whisker plot was used. A Box–Whisker plot displays the distribution of optimisation results of annual building energy consumption, based on twenty runs. A low median value (red line in the box) with small spread suggests a reliable algorithm in finding high-quality solutions in any experiment. A comparison between results in Figure 5.2 and Figure 5.3 shows that for both cities, the variability of ACOMV-M is very small. PSOHJ also performs well with the median value, which is close to ACOMV-M. By contrast, the spread of the optimisation results in ACOMV is larger than others. As can be seen for both cities, PSOHJ converged to some solutions that are relatively far from its median values (outliers).

Table 5-5: Optimisation results (**Bold** indicates the best found over all algorithms)

City		**PSOHJ** $(\mathbf{MJ/m^2/year})$	**ACOMV** $(\mathbf{MJ/m^2/year})$	**ACOMV-M** $(\mathbf{MJ/m^2/year})$
Brisbane	Solution (Median)	714.86	716.37	**714.69**
	Solution (Best)	714.74	715.36	**714.67**
Hobart	Solution (Median)	596.72	598.10	**596.67**
	Solution (Best)	596.67	597.40	**596.66**

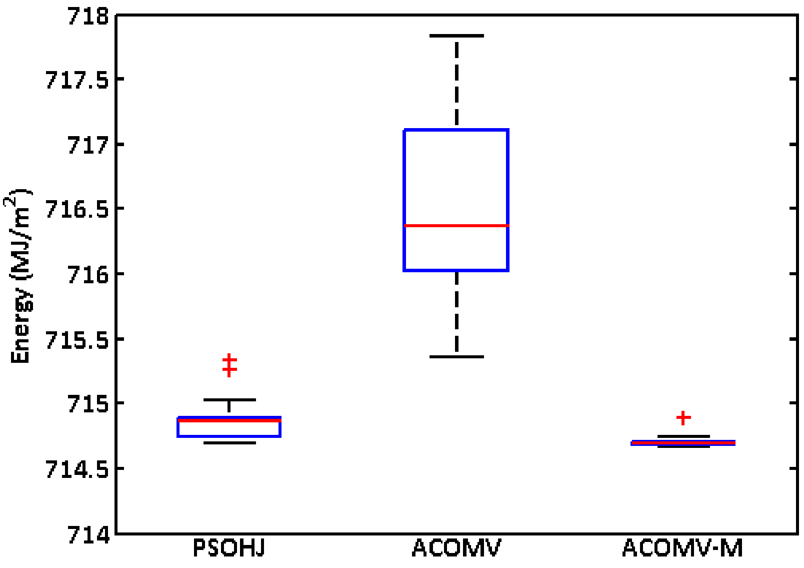

Figure 5.2 : Algorithm comparison with Box-Whisker plots for 20 runs for Brisbane

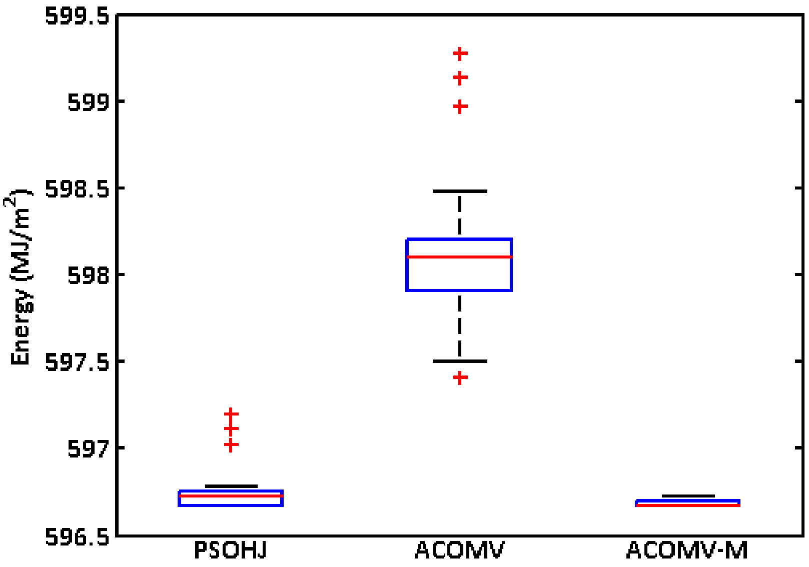

Figure 5.3 : Algorithm comparison with Box-Whisker plots for 20 runs for Hobart

To understand whether the difference between two algorithms is statistically significant, the Wilcoxon rank-sum test was applied. In this test, the null hypothesis (p-value > 0.05) means that there is no significant difference between two algorithms and the results may have been obtained by random chance. As shown in Table 5-6, the differences between ACOMV-M and other algorithms are indeed statistically significant.

Table 5-6: Wilcoxon rank-sum test results

City	ACOMV-M VS PSOHJ (*P*-value, h)	ACOMV-M VS ACOMV (*P*-value, h)	PSOHJ VS ACOMV (*P*-value, h)
Brisbane	(7.510e-06 ,1)	(6.795e-08 ,1)	(6.709e-08,1)
Hobart	(6.415e-04, 1)	(6.766e-08, 1)	(5.784e-08, 1)

The last comparison metric for optimisation algorithms is the convergence rate. In BOPs, the evaluation of objective function is frequently time-consuming. It is therefore essential that the number of building simulations is kept to a minimum, particularly when the algorithms' overall performance is very close in terms of the value of objective functions (as is the case here). Figure 5.4 and Figure 5.5 show an example of a convergence curve for a solution close to the median value among twenty runs. As can be seen for both cities, PSOHJ decreases gradually and stops after 3800 building simulations, and then HJ refines the PSO results, which is more noticeable for Brisbane (Figure 5.4). In contrast, the ACOMV algorithm falls rapidly at the initial iterations and then remains relatively unchanged and converges to a solution that is far from the best solution. ACOMV-M is similar to ACOMV and drops rapidly at the initial iterations and then it shrinks q automatically when no improvement is found after a predefined number to search locally. As can been seen, the refinements in the local search are considerable in both cities. This algorithm converges to a final solution earlier than other algorithms.

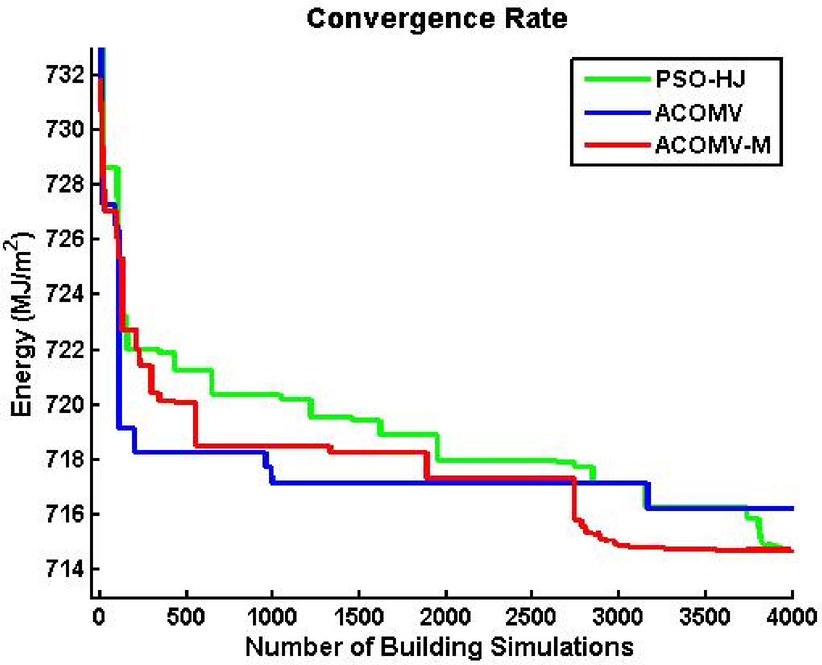

Figure 5.4: Convergence curve for the solution close to median for Brisbane

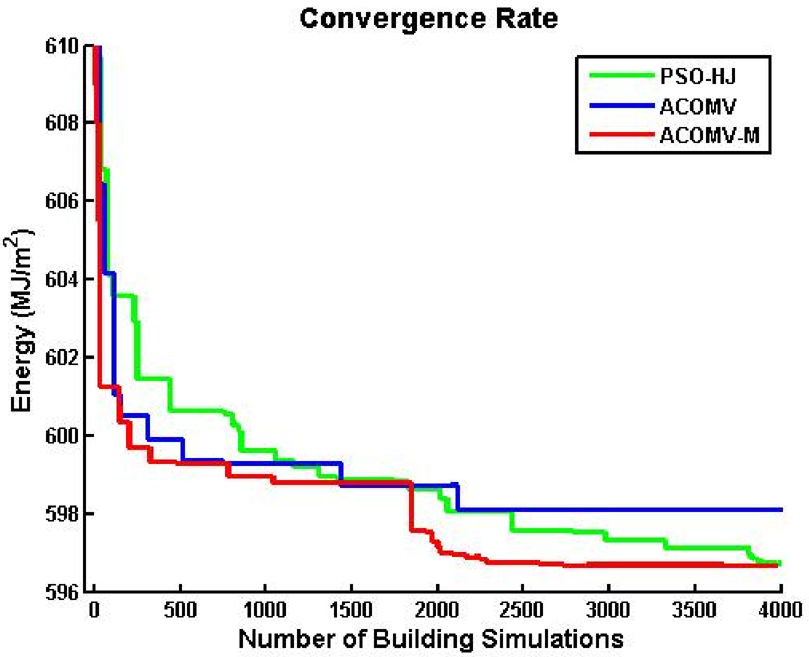

Figure 5.5: Convergence curve for the solution close to median for Hobart

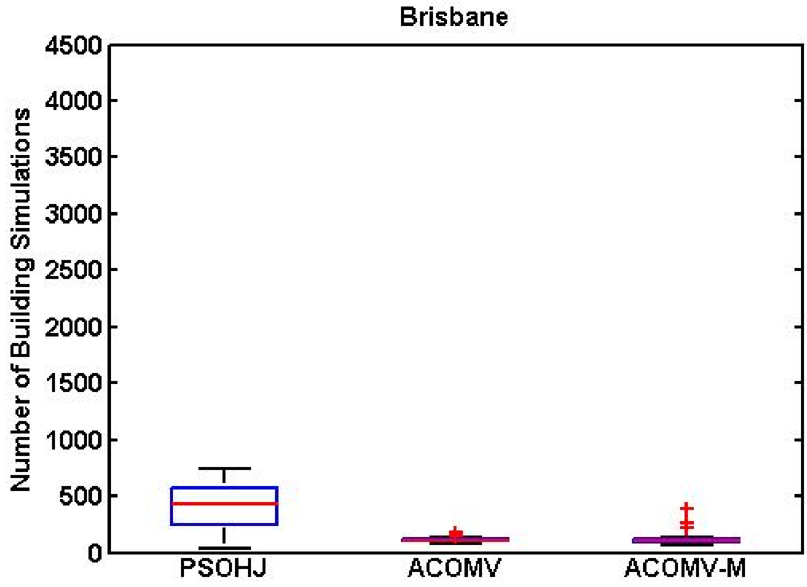

Figure 5.6: Number of simulations needed for each algorithm to achieve a solution within 1% of the best solution

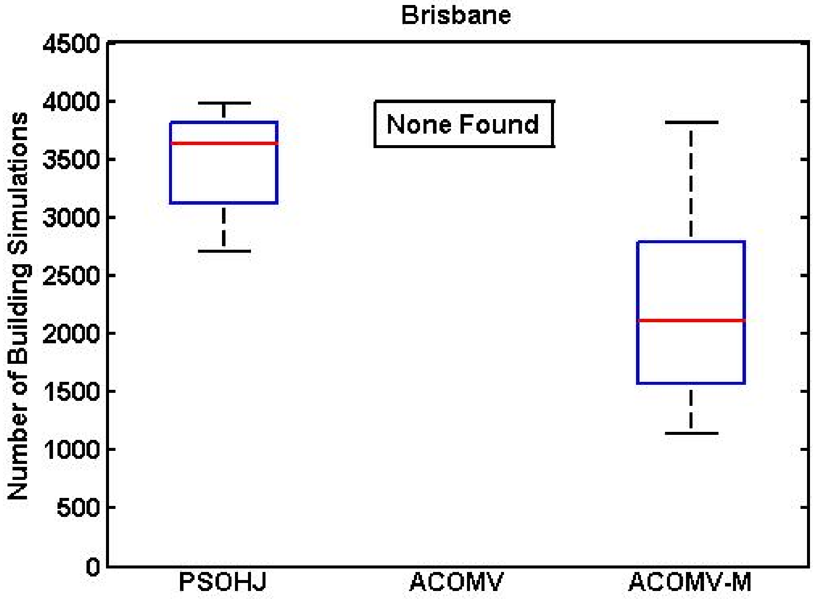

Figure 5.7: Number of simulations needed for each algorithm to achieve a solution within 0.1% of the best solution

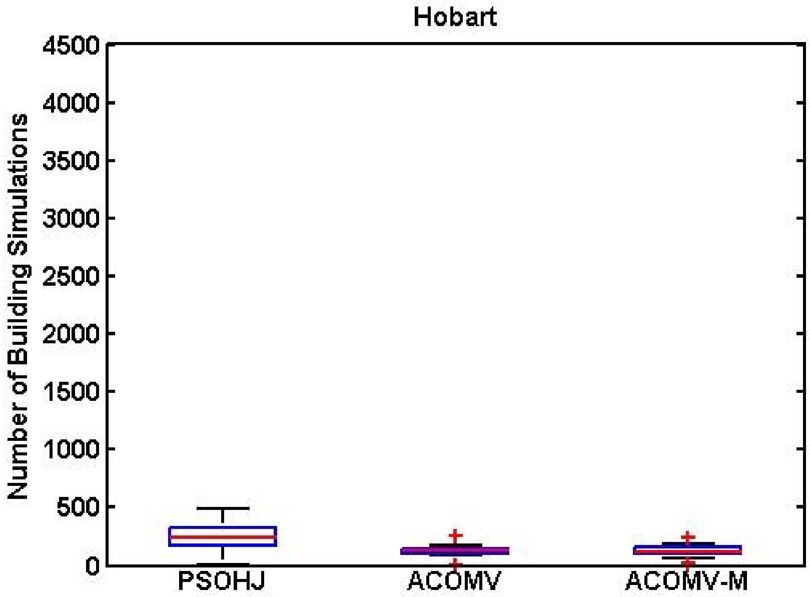

Fig. 5a

Figure 5.8:Number of simulations needed for each algorithm to achieve a solution within 1% of the best solution

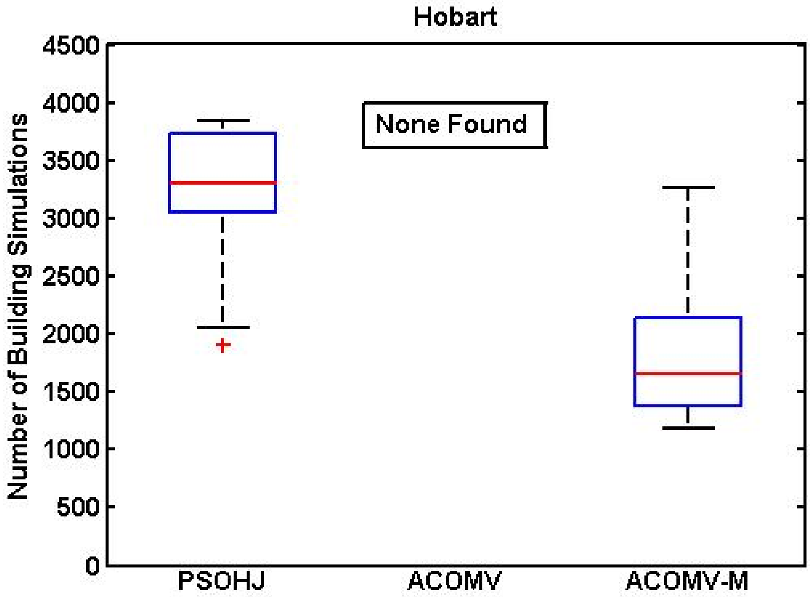

Figure 5.9: Number of simulations needed for each algorithm to achieve a solution within 0.1% of the best solution

Figure 5.6 shows the distribution of the number of building simulations when each algorithm achieved a solution within 1% of the best solution found (over all algorithms). As can be seen, all algorithms are able to find solutions within 1% with the small numbers of building simulations and with median values of 431.5, 108 and 105 for PSOHJ, ACOMV and ACOMV-M, respectively. However, both ACOMV and ACOMV-M algorithms are able to consistently find solutions faster than PSOHJ. Figure 5.7 shows the distribution of the number of simulations when each algorithm achieved a solution within 0.1% of the best solution found (over all algorithms). A comparison between median values shows that ACOMV-M (median value = 1653.5) is approximately two times faster than PSOHJ (median value = 3298.5). It should be noted that the ACOMV algorithm could not find any solutions 0.99% close to the best solution.

Figure 5.8 and Figure 5.9 show the distribution of the number of building simulations for Hobart when each algorithm achieved a solution within 1% and 0. 1% of the best solutions found (over all algorithms). Similar to Brisbane, this comparison shows that ACOMV-M has a faster convergence rate than other algorithms. Median values of distribution of the number of building simulations, when each algorithm achieved a solution within 1% and 0.1% of the best solutions, are presented in Table 5.7.

Table 5-7: Median values of distribution of the number of building simulations

Median values	Location	PSOHJ	ACOMV	ACOMV-M
within 1% of the best solution	Brisbane	431.5	108	105
within 0.1% of the best solution	Brisbane	3639.5	-	2110.5
within 1% of the best solution	Hobart	247.5	126.5	123
within 0.1% of the best solution	Hobart	3298.5	-	1653.5

5.6 Conclusion

In this chapter, firstly, to alleviate the computational cost of solving the BOPs with both continuous and discrete variables, a modified version of an ant colony optimisation algorithm called ACOMV-M was developed with the specific aim of localising the search in the later stages of optimisation. Then, this algorithm was applied to optimise nine variables (continuous and categorical) in a representative medium-size commercial building in both Brisbane and Hobart. A comparison between ACOMV, ACOMV-M and PSOHJ algorithms showed that ACOMV-M found solutions that are consistently slightly closer to the optimum. The Wilcoxon rank-sum test also statistically confirmed the better performance of ACOMV-M over other algorithms. In terms of convergence, both ACOMV algorithms converge to within 1% of the best solution faster than PSOHJ, however only ACOMV-M converges to within 0.1% faster than PSOHJ (approximately 50% fewer building simulations). After applying the optimisation method, up to 19% and 26% energy savings were achieved for Brisbane and Hobart, respectively.

Chapter 6: Uncertainty in Optimisation

6.1 Uncertainty in Building Simulation and Optimisation

In the vast majority of simulation/optimisation problems, building designers assume that building input parameters are deterministic (or perfectly known). However, in real building problems, especially at the early stages, parameters are often highly uncertain. These uncertainties may arise from different sources, including uncertainties in the thermophysical properties of construction materials and in weather data, lack of designers' knowledge of building occupancy, occupant behaviour and appliance loads, and uncontrolled infiltration rates [99, 100]). Due in part to this uncertainty, the simulated building and actual energy consumption may be quite different (i.e. the "performance gap" noted in many studies [103, 106]). In BOPs, this sensitivity to uncertain quantities implies that the "optimised" building may be far from the actual optimum.

The common methods to address uncertainty such as Monte Carlo simulation require probabilistic distributions of parameters that may not be available or representative, particularly in light of the fact that uncertainties may change during the lifetime of buildings [100]. In such cases, *scenario analysis* (i.e. analysing the behaviour of the building under a number of different specific building assumptions) may provide a complementary tool to enable uncertainty analysis when detailed distributional information is lacking [116].

Accordingly, in this chapter, the sensitivity of the optimised parameters to different simulation assumptions is first investigated in Section 6.2. It should be noted that the sensitivity of the optimised design for each scenario to optimisation variables and finding the robust optimised design for each scenario are an important subject which were investigated in many studies (see section 2.4) and was not studied here. In order to increase the robustness of optimised building (motivated by sensitivity study) a new formulation is then developed based on scenario-based, multi-objective optimisation in Section 6.3. Finally, the chapter conclusion is presented in Section 6.4.

6.2 Sensitivity of Optimised Building to Uncertain Parameters

In order to investigate the sensitivity of the optimised parameters to different simulation parameters, building Type B (stated in Chapter 3:) is simulated and optimised under three different scenarios: "base", "low" and "high" scenarios for two different climate zones, Brisbane and Hobart. Details of these scenarios are listed Table 6.1. The values of parameters for low and high scenarios were taken from previous studies that modelled building Type B [102]. As methods used for uncertainty analysis are often computationally expensive (see section 2.4), algorithms should be used that benefit from both accuracy and high convergence speed. Therefore, ACOM-M was used for optimisation as it has shown its suitability for BOPs in the previous chapter.

The optimisation objective function is the building annual end use energy consumption (Eq. 5.2). The nine variables applied to the case study in Chapter 5 were also selected here (Table 5.1 and Table 5.2). The building's characteristics and construction properties were detailed in Chapter 3:. However, two modifications including daylighting control and removal of temperature set back were added to this building before optimisation (the same as the case study stated in section 5.4).

Table 6-1: Base, high and low scenarios [102]

Parameter	Base Scenario	Low Scenario	High Scenario
Lighting (W/m^2)	15	9.3	21
Equipment (W/m^2)	15	7.5	20
Occupant (m^2/person)	10	50	5
Infiltration rate (ACH)	1	0.25	1.5

6.2.1 Results

Table 6.2 presents the results of simulation-based optimisation method using ACOMV-M for three scenarios for Brisbane and Hobart. As can be seen, different parameter sets were obtained in each scenario, showing that the optimised building design is highly sensitive to building simulation inputs. According to optimisation results for Brisbane, contrary to intuition, low insulation thicknesses are superior for energy consumption in the base and high scenarios. This is likely due to high internal loads in the building with a high scenario during daytime as well as Brisbane’s climate [152, 156]. With regard to shading size, in all scenarios the values near the maximum were chosen by the optimisation algorithm in the north face, while optimised values for other faces are different. The optimised window type was found to be double-glazed reflective windows in the north and south faces, and double-glazed tint windows with low emissivity in the west face in all scenarios.

For Hobart, regardless of scenarios, the maximum insulation thickness was selected by the optimisation algorithm. The optimised values of optimisation variables depend largely on building simulation assumptions and direction of building face. For example, for the North, South and West building faces, for a building with base scenario, the optimised window type is double pane tint windows with the low emissivity glass while for the building with low scenario, optimised window type is triple pane windows with low emissivity glass.

Table 6-2: Optimisation results in different scenarios

City	Scenario	Energy (MJ/ m^2/ year)	Insulation (cm)	Shading (N)	Shading (S)	Shading (E)	Shading (W)	Win type (N)	Win type (S)	Win Type (E)	Win type (W)
Brisbane	Base	714.67	2.6	1.172	1.166	0.457	0.511	6	6	8	9
	Low	310.82	10	1.169	1.169	0.511	1.199	6	6	6	9
	High	1004.2	1	1.168	0.750	0.224	0.506	6	6	11	9
Hobart	Base	596.66	10	1.169	0.479	0.511	0.506	9	9	11	9
	Low	287.35	10	0.844	0.041	0.511	1.200	11	11	9	11
	High	813.79	10	1.168	0.042	0.479	0.511	6	6	4	11

Table 6.3 shows the building annual energy consumption for all scenarios before and after optimisation. After applying the simulation-based optimisation method, the energy consumption of base, low and high scenarios reduced by 8.7%, 19.0% and 6.4% for Brisbane and 13.1%, 26.2% and 9.1% for Hobart, respectively. This table also shows the effect of inaccurate simulation inputs on saving energy obtained by the simulation-based optimisation method. For example, to simulate an underestimation of the internal loads, the optimised parameters for the low scenario are applied to a building whose internal loads are actually the base scenario. This leads to a reduction in energy savings from 13.1% to 11.9% in Hobart. In the extreme cases (i.e. considering the low scenario while building actual scenario is similar to the high

scenario) applying optimisation methods may reduce the building energy saving by 3.0 and 4.8 percentage points for Brisbane and Hobart, respectively.

Table 6-3: Building energy consumption before and after optimisation

	Location	**Brisbane**		**Hobart**	
		Energy (MJ/m^2/ year)	Energy Saving (compared to before optimisation)	Energy (MJ/m^2/ year)	Energy Saving (compared to before optimisation)
Base Scenario	Before optimisation	782.69	-	686.65	-
	After optimisation	714.67	8.7%	596.66	13.1%
	with optimised parameters of low scenario	720.62	7.9%	604.46	11.9%
	with optimised parameters of high scenario	**716.22**	8.5%	598.95	12.7%
Low Scenario	Before optimisation	383.71	-	389.28	-
	After optimisation	310.82	19%	287.35	26.2%
	with optimised parameters of base scenario	316.60	17.5%	294.95	24.2%
	with optimised parameters of high scenario	322.03	16%	305.79	21.4%
High Scenario	Before optimisation	1072.67	-	895.62	-
	After optimisation	1004.21	6.4%	813.79	9.1%
	with optimised parameters of base scenario	**1006.67**	6.1%	816.09	8.8%
	with optimised parameters of low scenario	1021.54	4.7%	827.95	7.5%

6.3 Increasing Robustness Using Multi-Objective Optimisation

In order to increase the robustness of optimised design to uncertainty in building simulation inputs (motivated by the sensitivity study), a new optimisation formulation was developed based on scenario-based multi-objective optimisation. This new formulation considers all three scenarios simultaneously during the optimisation and consequently increases the robustness of final solutions.

6.3.1 Problem Statement

The multi-objective optimisation can be generally stated as follows:

$$\min_{\mathbf{x}} \; [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_K(\mathbf{x})] \qquad 6.1$$
$$\text{subject to: } \mathbf{x} \in \mathbb{X} \subseteq \mathbb{R}^r \times \mathbb{V}^c$$

where $f(\cdot)$ is the objective function, K is the number of objective functions, $\mathbb{X}$ is the feasible space of independent design variables which is composed of continuous and categorical subspaces ($\mathbb{R}^r$ and $\mathbb{V}^c$, respectively). For both continuous and categorical variables in $\mathbf{x}$ the feasible design spaces are the same as described for Eq. 5.1.

The multi-objective optimisation problem considered here includes the building energy consumption under three different scenarios: a base (i.e. most likely) scenario $f_b(\mathbf{x})$, a low scenario $f_\ell(\mathbf{x})$, and a high scenario $f_h(\mathbf{x})$. The Weighted Sum Method (WSM) is used to scalarise the multi-objective optimisation problem. A weight ($w_i > 0$) is assigned to each objective function, which can be thought of as a relative probability of the scenario:

$$f(\mathbf{x}) = w_b f_b(\mathbf{x}) + w_l f_\ell(\mathbf{x}) + w_h f_h(\mathbf{x}) \qquad 6.2$$

$$w_b + w_\ell + w_h = 1 \qquad 6.3$$
$$w_\ell + w_h \leq w_b$$

where w_b, w_ℓ and w_h are the weights corresponding to each scenario and $f_b(\mathbf{x})$, $f_\ell(\mathbf{x})$ and $f_h(\mathbf{x})$ are calculated as follows:

$$f_i(\mathbf{x}) = E_c^i(\mathbf{x}) + E_f^i(\mathbf{x}) + E_\ell^i(\mathbf{x}) + E_p^i(\mathbf{x}) + E_h^i(\mathbf{x}) + E_m^i(\mathbf{x}) \qquad i = b, \ell, h \tag{6.4}$$

where for each scenario, E_c is the energy consumption for space cooling; E_f is the energy consumption of the fans; E_ℓ is the energy consumption of lighting; E_p is the energy consumption of pumps; E_h is the energy consumption for space heating and E_m is the energy consumption that includes interior equipment and heat rejection. It is noted that it is assumed here that the base (original) scenario represents the most likely scenario, with only combinations of weights that satisfy $w_\ell + w_h \leq w_b$ being considered, i.e. the base scenario is the most likely. This assumption comes from this fact that the according to the ABCB, base scenario is the typical scenario for commercial buildings (the most likely one ($w_b \geq 0.5$)) while there might be some deviations from the base scenario.

6.3.2 Robust Optimised Design Case Study

In this section, the results of the multi-objective optimisation using the Weighted Sum Method (WSM) are presented (Eq. 6.2). WSM was applied to convert the three-objective optimisation problem into a single-objective problem and ACOMV-M was applied to minimise the weighted objective. The weights are varied with a step size equal 0.1 to explore the effect of different scenario weights on the optimised building. For each combination of weights, five runs were conducted and the best run was selected. A limit of 9000 building simulations was selected for ACOMV-M algorithm, taking approximately 115 hours for each optimisation run with a high performance computing cluster.

Table 6.4 shows the multi-objective optimisation results for Brisbane. A comparison of objective function values between this table and Table 6.3 (i.e. single-objective) gives a sense for how much energy is “sacrificed” if a decision maker selects a solution with a higher degree of robustness with respect to possible changes in design scenarios. As can be seen in Table 6.4, it is possible to make significant energy gains in the base scenario without large sacrifices in the energy consumption in other scenarios. For example, consider the first row of Table 6.4 where the high and base scenarios are equally weighted ($w_h = w_\ell = 0.5$). Both the base and the high energy consumption are approximately $0.35\ (\mathrm{MJ/m^2/year})$ above their single-scenario optimised values from Table 6.3. One can compare this with the results in Table 6.3, where overestimating internal loads (first bold row of Table 6.3) will yield a building that has energy consumption $1.5\ (\mathrm{MJ/m^2/year})$ above the actual optimised value ($714.67\ \mathrm{MJ/m^2/year}$). Alternatively, if one underestimates the internal loads (second bold row in Table 6.3), the energy consumption is $2.46\ (\mathrm{MJ/m^2/year})$ above the true optimised ($1004.21\ \mathrm{MJ/m^2/year}$). By considering the weighted sum of the base and high scenarios, the resulting design has been designed to a compromise between the two scenarios, resulting in a lower energy sacrifice when the designer’s simulation assumptions are erroneous. On the other hand, the same example shows that the low scenario is quite far from the optimised value (albeit still lower than before optimisation) since it was not considered in the optimisation ($w_\ell = 0$). A similar analysis can be done for the other rows of Table 6.4. In general, higher robustness can be achieved by small sacrifices in the optimality of a building to any one scenario.

With regard to building configuration, by changing the weighting factors, different sets of optimised design variables were obtained, highlighting the importance of considering uncertainty in optimisation problems. Increasing the influence of high scenario (w_h) leads to a decrease in the insulation thickness, so that when the high scenario has the same importance as the base scenario ($w_b = w_h = 0.5$), minimum insulation thickness is required. Regarding shading size, different values were obtained in the south and east faces, while values remained constant in the north and west faces. Results show that approximately the maximum shading size in the north face is required. It is also observed that the window type is independent from the scenario in the north, south and west faces, while in the east face it depends heavily on the building scenario weights, emphasising the sensitivity of east-facing windows to simulation assumptions.

Table 6.5 shows the multi-objective optimisation results for Hobart. As can be seen in this table, in contrast to Brisbane, maximum insulation thickness is required for all combinations of scenarios. The shading size is almost constant (approximately half a metre) in the east and west faces. On the other hand, different values were selected for shading size in the north and south faces. As shown, the optimisation algorithm found large values for shading size in the north face and small values for the south face, which confirms rule-of-thumb design guidelines. In southern cities in Australia, windows facing south receive less direct solar heat gain as they are frequently under the shadow due to angle of the sun in the south hemisphere. Therefore, they require smaller overhangs as they tend to benefit from the daylighting. With regard to windows, type 9 is required for the south and west faces for all cases, except when $w_b = w_h = 0.5$. However, in the east and north faces, window type depends on the scenario weights.

A comparison of objective function values between this Table 6.5 and Table 6.3 shows the amount of energy that is sacrificed to select a robust solution with respect to probable changes in simulation scenarios. For instance, when there is an equal weight for the base and low scenarios ($w_b = w_\ell$), a compromise solution was obtained by sacrificing approximately 2 $(\mathrm{MJ/m^2/year})$ and 3 $(\mathrm{MJ/m^2/year})$ energy of building with base and low scenarios, respectively.

Table 6-4: Multi-objective optimisation results for Brisbane

w_b	w_ℓ	w_h	Scenarios Energy $(\mathrm{MJ/m^2/year})$			Insulation (cm)	Shading (N)	Shading (S)	Shading (E)	Shading (W)	Win type (N)	Win type (S)	Win type (E)	Win type (W)
			Base	Low	High									
0.5	0.0	0.5	715.00	320.13	1004.60	1	1.17	0.75	0.24	0.51	6	6	9	9
0.5	0.1	0.4	715.22	318.23	1004.92	1	1.18	0.75	0.51	0.51	6	6	8	9
0.5	0.2	0.3	714.70	314.55	1007.48	4	1.18	1.17	0.48	0.51	6	6	8	9
0.5	0.3	0.2	714.97	312.14	1009.17	10	1.17	1.16	0.49	0.51	6	6	11	9
0.5	0.4	0.1	714.96	312.04	1009.21	10	1.17	1.17	0.49	0.51	6	6	11	9
0.5	0.5	0.0	714.98	312.03	1009.31	10	1.17	1.17	0.49	0.51	6	6	11	9
0.6	0.0	0.4	714.93	320.29	1004.80	1	1.18	1.17	0.24	0.51	6	6	9	9
0.6	0.1	0.3	714.92	319.64	1004.84	1	1.18	1.17	0.24	0.51	6	6	9	9
0.6	0.2	0.2	714.71	313.82	1007.91	5	1.17	1.17	0.49	0.51	6	6	8	9
0.6	0.3	0.1	714.97	312.03	1009.22	10	1.18	1.16	0.49	0.51	6	6	11	9
0.6	0.4	0.0	714.97	312.03	1009.31	10	1.17	1.17	0.49	0.51	6	6	11	9
0.7	0.0	0.3	714.97	320.13	1004.64	1	1.17	0.75	0.24	0.51	6	6	9	9
0.7	0.1	0.2	714.70	315.27	1007.06	3	1.17	0.75	0.48	0.51	6	6	8	9
0.7	0.2	0.1	714.81	313.02	1008.49	7	1.18	1.17	0.51	0.51	6	6	8	9
0.7	0.3	0.0	714.97	312.02	1009.31	10	1.17	1.17	0.49	0.51	6	6	11	9
0.8	0.0	0.2	714.71	319.62	1006.17	2	1.17	0.75	0.45	0.51	6	6	8	9
0.8	0.1	0.1	714.76	314.04	1007.81	4	1.17	1.16	0.47	0.51	6	6	11	9
0.8	0.2	0.0	714.96	312.06	1010.04	10	1.18	1.17	0.49	0.51	6	6	11	9
0.9	0.0	0.1	714.70	319.50	1006.29	2	1.18	1.17	0.45	0.51	6	6	8	9
0.9	0.1	0.0	714.71	313.85	1007.96	5	1.17	1.17	0.49	0.51	6	6	8	9

Table 6-5: Multi-objective optimisation results for Hobart

w_b	w_ℓ	w_h	Energy Scenarios $(\mathrm{MJ/m^2/year})$			Insula-tion (cm)	Shading (N)	Shading (S)	Shading (E)	Shading (W)	Win type (N)	Win type (S)	Win type (E)	Win type (W)
			Base	Low	High									
0.5	0.0	0.5	597.33	295.03	814.21	10	1.17	0.48	0.48	0.51	9	9	8	11
0.5	0.1	0.4	597.02	295.11	814.63	10	1.17	0.29	0.48	0.51	9	9	8	9
0.5	0.2	0.3	597.04	295.03	814.65	10	1.15	0.29	0.48	0.51	9	9	8	9
0.5	0.3	0.2	597.64	291.41	817.04	10	1.15	0.29	0.51	0.51	11	9	11	9
0.5	0.4	0.1	597.85	290.90	817.43	10	1.02	0.29	0.51	0.51	11	9	11	9
0.5	0.5	0.0	598.27	290.42	817.95	10	0.88	0.15	0.51	0.51	11	9	11	9
0.6	0.0	0.4	597.00	295.31	814.63	10	1.17	0.48	0.48	0.51	9	9	8	9
0.6	0.1	0.3	596.99	295.26	814.63	10	1.17	0.48	0.48	0.51	9	9	8	9
0.6	0.2	0.2	596.80	294.39	815.82	10	1.09	0.29	0.48	0.51	9	9	8	9
0.6	0.3	0.1	597.68	291.30	817.16	10	1.13	0.28	0.51	0.51	11	9	11	9
0.6	0.4	0.0	597.90	290.83	817.53	10	0.99	0.29	0.51	0.51	11	9	11	9
0.7	0.0	0.3	596.99	295.31	814.62	10	1.17	0.48	0.48	0.51	9	9	8	9
0.7	0.1	0.2	596.67	294.99	815.68	10	1.17	0.48	0.51	0.51	9	9	11	9
0.7	0.2	0.1	596.80	294.34	815.84	10	1.07	0.29	0.51	0.51	9	9	11	9
0.7	0.3	0.0	597.72	291.18	817.61	10	1.07	0.48	0.51	0.51	11	9	11	9
0.8	0.0	0.2	596.68	295.05	815.68	10	1.17	0.48	0.51	0.51	9	9	11	9
0.8	0.1	0.1	596.68	294.99	815.68	10	1.17	0.48	0.51	0.51	9	9	11	9
0.8	0.2	0.0	596.85	294.13	816.01	10	1.02	0.29	0.51	0.51	9	9	11	9
0.9	0.0	0.1	596.67	295.05	815.72	10	1.17	0.48	0.51	0.51	9	9	11	9
0.9	0.1	0.0	596.67	294.98	816.07	10	1.17	0.48	0.51	0.51	9	9	11	9

6.4 Conclusion

In this chapter, firstly the sensitivity of building-optimised parameters to building simulation inputs was examined for a representative medium-size commercial building in both Brisbane and Hobart. Nine variables, including both continuous and categorical variables were used, and simulation-based optimisation using the ACOMV-M algorithm was utilised as an optimisation method. The results showed that under different simulation inputs, the optimised parameters may vary significantly. Overestimation or underestimation of simulation assumptions (i.e. lighting and equipment loads, occupant density and infiltration rates) can reduce energy savings obtained by the simulation-based optimisation method up to 4.8 percentage points if the base assumptions are used in optimisation.

Secondly, using the new methodology based on multi-objective optimisation enables identification of sensitive (and insensitive) design parameters with respect to the variations of design scenarios. Additionally, the results show that small sacrifices in the optimality of a building to any one scenario can result in significantly robust solutions across all scenarios.

Chapter 7: Surrogate-based Optimisation

7.1 Overview

In this chapter, a new surrogate-based optimisation method called Surrogate-based Optimisation using Active Learning (SOAL) was developed and its results were compared to conventional surrogate-based optimisation and simulation-based optimisation methods. This chapter is structured as follows: Section 7.2 discusses artificial neural networks, which will be used to construct surrogate models; Section 7.3 details a new active sampling method; Section 7.4 details a new surrogate-based optimisation method; finally, Section 7.5 presents the results followed by a conclusion in Section 7.6.

7.2 Artificial Neural Network

Artificial neural networks (ANNs) have been selected to construct a surrogate model, as the literature (see Section 2.3) has found they perform well in both building energy prediction and optimisation problems. ANNs are computer-learning models, which were inspired by biological neural networks, that mimic the learning process of the human brain [67]. The first computational model of neural networks was introduced in [157]. An ANN is a network of artificial neurons (also known as artificial nodes) that seeks a relationship between the input parameters and outputs without any information about the system and only by analysing previously recorded data. The architecture of a neuron is depicted in Figure 7.1.

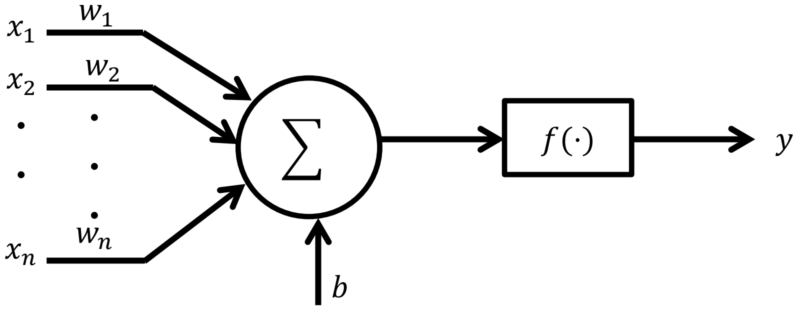

Figure 7.1: Model of a neuron

As shown, a neuron receives inputs, which are multiplied by the connection weights, and produces an output signal by applying an activation function (transfer function) to the weighted sum of its inputs. Activation functions are frequently non-linear functions such as the sigmoid function or hyperbolic tangent function, which enable neurons to model complex non-linear functions. The output of a neuron (y) can be written as follows:

$$y = f\left(\sum_{i=1}^{n} w_i\, x_i + b\right) \qquad 7.1$$

where x_i represents the ith input of the neuron, w_i is the weight associate with ith input and b is the bias. The most common ANN is the feed-forward Multi-Layer Perceptron (MLP), which consists of a set of neurons in different layers including one input layer, one or more hidden layers and one output layer [72, 158]. It has been shown that an MLP including a single hidden layer with an appropriate number of neurons is able to approximate any function with arbitrary accuracy (i.e. it is a universal approximator) [158-161]. A schematic of neural network architecture is shown in Figure 7.2. The input layer consists of several neurons, which receive input parameters, and neurons in the output layer provide the outputs of the network. Each single neuron is connected to all other neurons in the previous layer through *weights*. In order to identify the values of weights, the network is trained using historical data, and the optimal value of the weights is found by minimising the Mean-Squared Error (MSE) output of the network (predicted values) and actual (desired) values. Back-Propagation is the most widely used training method for the neural networks [20, 65], which is essentially a gradient descent method that seeks a (local) minimum of the MSE.

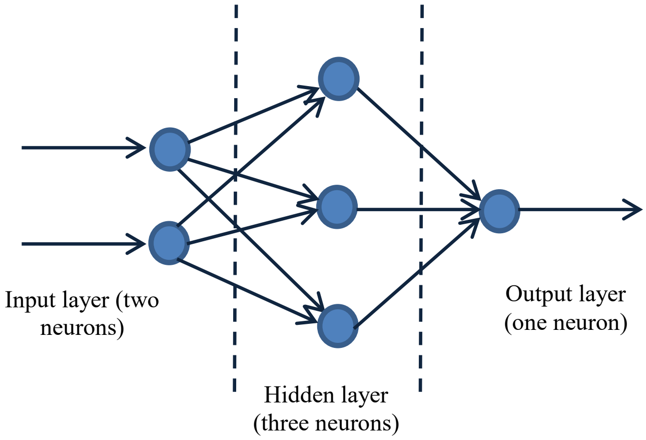
Input layer (two neurons)
Hidden layer (three neurons)
Output layer (one neuron)

Figure 7.2: Schematic diagram of a multilayer neural network

A key parameter in the performance of ANNs is the number of neurons in hidden layers. The number of neurons depends strongly on the problem and should be properly selected. Too many neurons in the hidden layers can lead to overfitting. This occurs when the network is fit too closely to the training data and the error on the training set is very small, while the error of the network on unseen data is large. This means that the neural network does not generalise well to new samples. On the other hand, if the number of neurons in the hidden layers is too few, the model fails to capture the trend of the data (under-fitting). Therefore, the network performs poorly on both training and new data. Thus, finding a balance between the fitting performance and the generalisation performance is essentially a question of determining the number of hidden neurons.

Thus, in order to use ANNs as a surrogate model, the training of the weights and the method for determining of the number of hidden neurons must be specified. These issues will be discussed in the next sections.

7.2.1 Neural Network Training Algorithm

The neural network training process aims to optimise the values of weights and biases in order to minimise network error (difference between network outputs (network predictions) and desired outputs). The Levenberg-Marquardt (LM) algorithm [162, 163] is an efficient optimisation algorithm that has been widely used for training ANNs [164] and is highly recommended as a first-choice algorithm for supervised learning problems [165]. The LM algorithm is a combination of the Gauss–Newton algorithm and the steepest descent method so that it benefits from the stability of the steepest descent method and the speed advantage of the Gauss–Newton algorithm [72, 164, 166]. In this algorithm the weights are updated as follows:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \left(\mathbf{J}_k^{\mathrm{T}} \mathbf{J}_k + \mu\, \mathbf{I}\right)^{-1} \mathbf{J}_{\mathrm{k}}\, \mathbf{e}_{\mathrm{k}} \qquad 7.2$$

where $\mathbf{w}$ is the weight vector, $\mathbf{J}$ is the Jacobian matrix containing first derivatives of the network errors with respect to the weights, $\mathbf{e}$ is the vector of network errors and $\boldsymbol{I}$ is the identity matrix,. The term ($\mathbf{J}^{\mathrm{T}}\mathbf{J}$) is the approximation of Hessian matrix from Gauss–Newton Algorithm and μ is a combination coefficient. When μ is very small, the Levenberg Marquardt algorithm performs like the Gauss–Newton algorithm. In contrast, when μ is very large, the Levenberg Marquardt algorithm performs like the gradient descent algorithm with a small step size.

7.2.1.1 Early Stopping and Regularization

Neural network generalisation is the prediction capability of a network over new (unseen) data. A network that performs well on training data may perform poorly on new data due to the well-known issue of overfitting, which can occur due to 1) fitting too complex of a model (too many hidden neurons) or 2) carrying out too many iterations of the iterative method used to train the weights [68, 72, 77]. While cross validation is used to prevent 1), 2) is typically mitigated by either early stopping or regularisation.

In early stopping, the training process is terminated before the algorithm's convergence criteria are satisfied. The labelled data is split into two subsets: *training* and *validation*. The training subset is used for updating the network weights and biases, while the validation subset is used to determine the onset of overfitting. During the initial iterations of the training process, both the training and validation errors decrease. However, when the network starts overfitting, the network's prediction error over the validation subset starts increasing, at which time the training is terminated and the network with the minimum validation error is selected.

A second method to overcome the overfitting issue is Bayesian regularisation. The idea of regularisation is to add a regularisation term to the network performance function with the aim of penalising the large values of weights. Typically, the performance function for training feed-forward neural networks is the MSE of the network, which can be written as follows:

$$F_o = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(t_i - y_i)^2 \qquad 7.3$$

where y_i is the output of the neural network for ith input, t_i is the desired value associated with for ith input, and N is the number of training samples. In this method, in order to improve the generalisation capability, a term indicating the "size" of the network weights and biases is added to the performance function, which can be written as follows:

$$F_r = \gamma F_w + (1 - \gamma) F_o \tag{7.4}$$

where F_r is the modified performance function in the regularisation method, γ is the performance ratio, F_w is the mean of the sum of squares of the network weights and biases, which is calculated as follows:

$$F_w = \frac{1}{N}\sum_{i=1}^{N} w_i^2 \tag{7.5}$$

The modified performance function (Eq.7.4) aims to find a compromise between finding small weights and minimising the original performance function (Eq. 7.3). In modified performance function, smaller values of weights and biases are preferred compared to original cost function, and therefore the network response is smoother and less likely to overfit [165].

Both early stopping and regularisation methods can significantly improve network generalisation when they are applied properly. However, Bayesian regularisation provides better generalisation performance for the small data set since it does not require a validation subset and therefore can use all of the training data [165].

7.2.1.2 Cross Validation

One well-established method for determining the appropriate number of hidden neurons is *cross validation,* where some of the training data is removed from the training set and used to assess the generalisation performance of the model [167]. For k-fold cross-validation, training data are divided into k subsets of (approximately) equal size and then the network is trained k times so that each time, one of the subsets is left out from training data and used as test data, and the remaining $(k - 1)$ data sets are used for training the network. The model performance is then expressed as the average prediction (generalisation) error over all k test folds. The optimal number of hidden neurons can be found by selecting the number of neurons that result in the lowest average prediction error. Thus, the K-fold cross validation process is repeated until the model generalisation error stops improving for specific number of iterations. Finally, the model with the minimum prediction error (i.e. maximum generalisation performance) is chosen.

The key parameter in this method is the value of K which should be selected appropriately [168]. Although there are no generally accepted mathematical formula for determining the number of neurons in the hidden layer [158], Kohavi [169] investigated the effect of different values of K on many real-world datasets and their results showed that the cross validation method with ten folds is suitable for model validation.

7.3 Sample Selection Method

The typical sample selection method in building performance and optimisation problems is random sampling. In this method, sample points are randomly selected to train the surrogate model. Due to the random selection, some samples may contain less information and not be the representative of the whole design space. Therefore, more samples (and higher computational cost) are required to train the surrogate model to reach the desired prediction accuracy.

A new sample selection method is developed in this research to improve the efficiency of the surrogate-based optimisation method. The proposed method aims to improve the prediction of a surrogate model by selecting the most representative and informative samples (samples with high uncertainty) only in the regions where the predicted energies are low (around the local minima) to focus building simulations in areas of the parameter space that have high potential to be near local minima.

Let $L = \{x_n, y_n\}_{n=1}^{N}$ denote the initial training dataset composed N labeled samples and $U = \{\hat{x}_m\}_{m=1}^{M}$ denote the pool of M unlabeled samples where $M > N$. In order to generate a set of samples, Latin Hypercube Sampling (LHS) is used to generate both L and U to ensure efficient coverage of the entire parameter space [170]. In this method, the range of each design variable is divided into n non-overlapping intervals with equal probability. A sample is then selected randomly on each interval of every design variable.

In order to select the most informative unlabeled samples for labeling, a *committee* consisting of P surrogate models is built using the initial labeled dataset (L) with different weight initialisations (so that each ANN may achieve a different local optimum). Each surrogate model predicts the label of every unlabeled sample point in the unlabeled pool of data set (U). Let the predicted values by the pth committee member for $\hat{x}_m$ be $\hat{y}_m^p$. Then, the mean and variance of predicted values for $\hat{x}_m$ over all P committee members may be calculated as follows:

$$\bar{y}_m = \frac{1}{P}\sum_{p=1}^{P}(\hat{y}_m^p) \tag{7.6}$$

$$v_m = \frac{1}{P}\sum_{p=1}^{P}(\hat{y}_m^p - \bar{y}_m) \tag{7.7}$$

In this method, the variance of samples provides an idea of the level of disagreement between surrogate models. Samples with higher variance are those which are more uncertain and could add more information to improve the surrogate model prediction accuracy. Thus, unlabeled sample points are then sorted from the highest to the lowest variance and the first k unlabeled samples $(k \leq M)$ with the highest variances are good candidates for new samples to label. However, it is also highly desirable to select samples that are high predicted quality (i.e. low predicted energy) since these samples are more likely to be near the optimal parameters. Thus, a condition on the quality of the selected samples is set as well. Let $\overline{\boldsymbol{Y}} = \{\bar{y}_1, \bar{y}_2, \dots, \bar{y}_M\}$ be the vector of predicted mean value of unlabeled samples. Only the samples satisfying

$$\bar{y}_m < Pr_j\,(\overline{\boldsymbol{Y}}) \qquad m = 1,2,\dots,k \tag{7.8}$$

are selected for labelling. In the above equation, Pr_j is a function which returns the jth percentile of unlabeled samples in the pool and decreases at each optimisation iteration. This selection strategy thus leads to an improvement of the accuracy of the surrogate model in promising regions (regions with low energy).

7.4 Proposed Surrogate Based Optimization Method

A new surrogate-based optimisation method called Surrogate-based Optimisation using Active Learning (SOAL) is developed in this section. The flowchart of this method is illustrated in Figure 7.3.

In this method, first an initial surrogate model is constructed with a small number of labelled samples (initial training dataset) generated by LHS. In the next step, in order to identify the best architecture of the network (i.e. number of hidden neurons), K-fold cross validation is applied. The network is then trained P times with different random initialisations to build a committee of networks consisting of P surrogate models. Each of the P models will result in a network with different accuracy.

At this point $\hat{P} \in [1, P]$ surrogate models are optimised. Two variants of the algorithm are used:

1. $\hat{P} = 1$. The best surrogate model (i.e. surrogate model with maximum generalisation performance) in the committee is used for optimisation in each iteration.
2. $\hat{P} = P$. All members of committee are optimised in each iteration.

Since the optimisation process does not require any further building simulations (only evaluation of the surrogate model), each optimisation is much faster than the software-in-the-loop approach. The ACOR algorithm detailed in 4.4.1 is used for optimisation of surrogate model(s) and $\hat{P}$ optimised solutions are stored in a library for future labelling via building simulation. In each iteration, the objective function of corresponding optimised solution(s) is calculated by EnergyPlus and compared with its value from previous iterations and then the library is updated with the smaller value. The solution stored in the library represents the best solution found so far.

If the stopping criterion (i.e. maximum number of iterations) is not satisfied, new samples are generated for subsequent labelling by the building simulation to refine the surrogate model. Two methods are used to generate the next (k) samples:

In the first method, $\hat{P}$ optimised solutions obtained by the committee of surrogate model(s) in the current iteration are added to the training data set (dataset of building simulation results) for the next iteration. These samples are likely close to local minima and hence they have the potential to improve the model prediction accuracy in promising regions (resulting in local refinement).

In the second method, the proposed sample selection method stated in section 7.3 is used to generate the remaining $k - \hat{P}$ new samples. Accordingly, a pool of U unlabeled samples is generated using the LHS method. The variance and mean of each sample is then calculated using Eq. 7.6 and Eq. 7.6, and sorted from the highest to lowest variance. The first $k - \hat{P}$ samples, which satisfy Eq. 7.8, are selected to be labeled by EnergyPlus and then are added to the training dataset. These samples are removed from the pool of unlabeled samples. This process is repeated until the stopping criterion is satisfied.

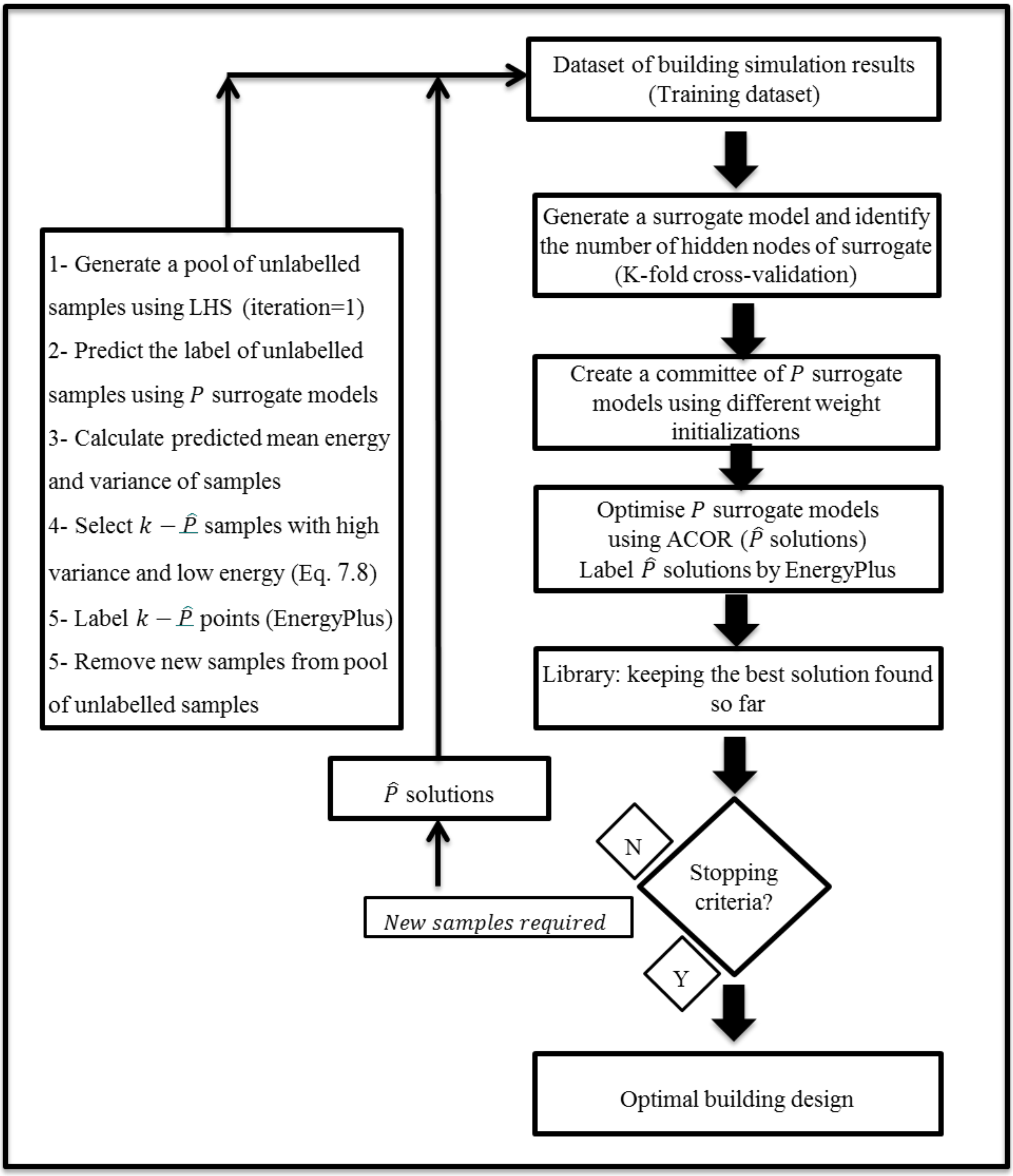
Dataset of building simulation results (Training dataset)
Generate a surrogate model and identify the number of hidden nodes of surrogate (K-fold cross-validation)
Create a committee of P surrogate models using different weight initializations
Optimise P surrogate models using ACOR ($\hat{P}$ solutions)
Label $\hat{P}$ solutions by EnergyPlus
Library: keeping the best solution found so far
Stopping criteria?
N
Y
Optimal building design
$\hat{P}$ solutions
New samples required
1- Generate a pool of unlabelled samples using LHS (iteration=1)
2- Predict the label of unlabelled samples using P surrogate models
3- Calculate predicted mean energy and variance of samples
4- Select $k - \hat{P}$ samples with high variance and low energy (Eq. 7.8)
5- Label $k - \hat{P}$ points (EnergyPlus)
5- Remove new samples from pool of unlabelled samples

Figure 7.3: Flowchart of surrogate-based optimisation using active learning

7.5 Results

The SOAL method was applied to building Type B for two cities: Brisbane and Melbourne. The building's characteristics and construction properties were detailed in Table 3.3 and Table 3.4 (Chapter 3:). Two modifications were added to this building before optimisation (similar to the case study Section 5.4). First, daylighting control for each perimeter zone was added. Secondly, temperature set back was removed.

The objective function is to minimise the annual energy consumption of the building, which was stated at Section 4.2 with respect to 15 variables listed in Table 7.2. To conduct surrogate model-based optimisation, a standard feed-forward multi-layer perceptron ANN with three layers (input, hidden, and output) was used for the surrogate model. The sigmoid function was used as the activation function and all input data were normalised between [0, 1]. Latin hypercube sampling method was used to generate a pool of unlabelled samples with 15000 sample points. A MATLAB code was developed to run EnergyPlus automatically and control the whole optimisation process, including network training and optimisation algorithm.

The Levenberg–Marquardt back-propagation algorithm with Bayesian regularisation was used to train the network. The algorithm parameters were selected based on recommendations in [165] and listed in Table 7.3. Once the training process was completed, the ACOR algorithm was applied to optimise the surrogate model(s). The initial surrogate model was built using 50 training samples. These initial samples were labelled by EnergyPlus. In other words, EnergyPlus calculates annual end-use energy consumption (Eq. 4.2) associated with each sample (i.e. each sample includes a set of fifteen variables shown in Table 7.2 which their values are selected through the Latin Hypercube method). The committee of surrogate models contains five members ($P = 5$) with different initialisations. In each iteration, 50 new samples were added to the training dataset ($k = 50$). The values of function Pr_j were calculated based on Table 7.1.

Table 7-1: Percentiles for the (P)SOAL method

Iteration number	1	2	3	4	5	6	7	8	9	10	11	12	⋯	40
Pr_j (Percentile)	50	45	40	35	30	25	20	15	10	5	1	1	⋯	1

To conduct the simulation-based optimisation, PSOIW (using GenOpt software) and ACOR algorithms were directly connected to EnergyPlus. The algorithms' parameters were chosen based on the recommendations of previous studies [44, 151] and are as listed in Table 4.3 and Table 4.4 (Chapter 4:).

Table 7-2: Optimisation variables

Variables	Description	Initial Value	Variable Range
x_1	Roof emissivity	0.7	[0.5-0.9]
x_2	Roof solar absorptance	0.7	[0.3-0.85]
x_3	Wall insulation (cm)	4.5	[1- 10]
x_4	Wall solar absorptance	0.7	[0.3-0.9]
x_5	North window height (m)	1.35	[0.5-1.5]
x_6	South window height (m)	1.35	[0.5-1.5]
x_7	East window height (m)	0.54	[0.5-1.5]
x_8	West window height (m)	0.54	[0.5-1.5]
x_9	North overhang depth (m)	-	[0-1.5]
x_{10}	South overhang depth (m)	-	[0-1.5]
x_{11}	East overhang depth (m)	-	[0-1.5]
x_{12}	West overhang depth (m)	-	[0-1.5]
x_{13}	Heating setpoint (°C)	20	[18-22]
x_{14}	Cooling setpoint (°C)	24	[23-27]
x_{15}	Building orientation (degree)	0	[0-45]

Table 7-3: Parameters of Levenberg–Marquardt with Bayesian regularisation

Parameters	Value
Maximum number of epochs to train	2000
Marquardt adjustment parameter	0.005
Decrease factor for mu	0.1
Increase factor for mu	10
Maximum value for mu	1e10
Minimum performance gradient	1e-7

Fifteen optimisation runs were conducted for each optimisation algorithm using QUT's High Performance Computing (HPC) cluster, since 2000 building simulations were required for each run. The time required for 2000 building simulations with EnergyPlus 8.1.0 is approximately 25 hours. In surrogate-based optimisation methods, training time depends on the number of sample points used for training, which took less than ten minutes for 2000 samples.

7.5.1 Network configuration identification

K-fold cross validation was used to determine the optimal number of neurons in the hidden layer. A 10-fold cross validation is selected in this research, which has been recommended by previous studies [168, 169]. The performance of the network is then calculated as the average accuracy of the networks over 10 folds. MSE is used to evaluate the performance of the each network.

Figure 7.4 shows an example of the MSE performance for 10-fold cross validation for an ANN that was trained using 500 samples for Brisbane. As can be seen, the minimum MSE performance was achieved for a network with 6 neurons in the hidden layer. During the optimisation process, new samples were added to the training data and this process was applied to determine the optimal configuration of the network in each iteration. Similarly, Figure 7.5 shows the MSE performance for 10-fold cross validation for Melbourne. As shown, the optimal configuration was achieved for the network when the number of neurons in the hidden layer is 8.

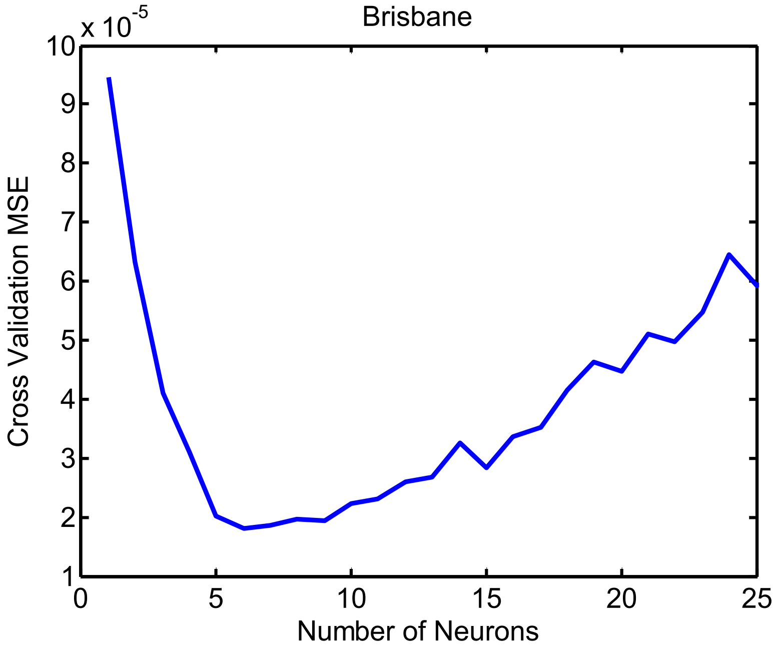

Figure 7.4: Cross validation MSE for Brisbane (averaged over 10 folds)

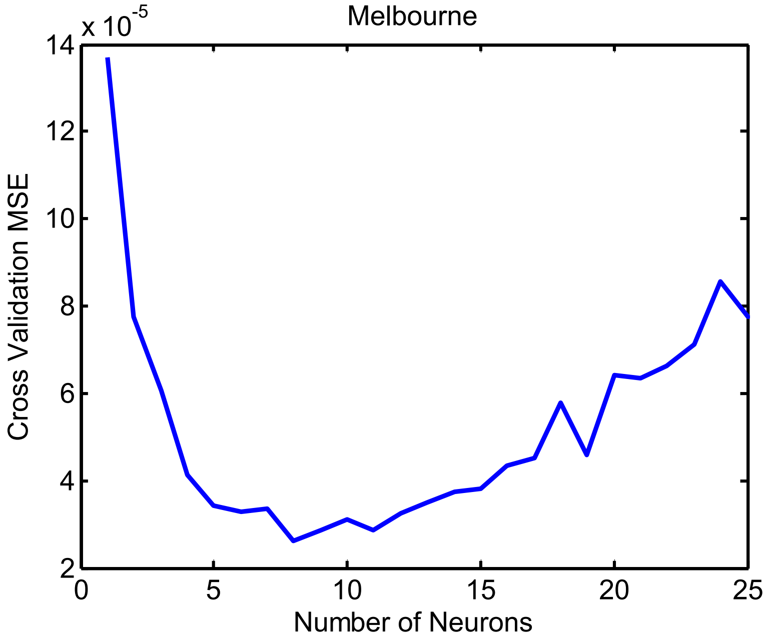

Figure 7.5: Cross validation MSE for Melbourne (averaged over 10 folds)

7.5.2 Optimisation Results

Figure 7.6 shows the best solution found as a function of the number of building simulations for Brisbane. This figure compares the results of five different methods, including two software-in-the-loop methods (PSOIW and ACOR in the figure), and three surrogate-based optimisation methods using different sampling strategies: random sampling, SOAL when $\hat{P} = 1$ and SOAL when $\hat{P} = 5$ (P-SOAL). In this figure, the median over the 15 optimisation runs are presented. A comparison of surrogate-based optimisation methods shows that the P-SOAL method performs the best and both active sampling methods outperform random sampling. This figure also shows that all surrogate-based optimisation methods perform better than software-in-the-loop with PSOIW algorithm. A comparison between surrogate-based optimisation methods and software-in-the-loop with ACOR shows all surrogate-based optimisation methods are able to find better solutions at the early stages of optimisation while after approximately 1000 building simulations both SOAL and P-SOAL remain competitive to the ACOR algorithm. As can be seen, ACOR performs slightly better than SOAL and P-SOAL after 1000 building simulations. The reason is likely due to the fact that surrogate model approximation is not exact; however, from an energy point of view these differences are quite small as a percentage of building energy consumption.

Figure 7.7 shows the median value of results of convergence curve for fifteen runs for Melbourne. Similar to results for Brisbane, all surrogate-based optimisation methods outperform PSOIW, and P-SOAL performs the best among surrogate model-based optimisation methods; it outperforms ACOR at the early stages of optimisation and it remains a competitive method to ACOR.

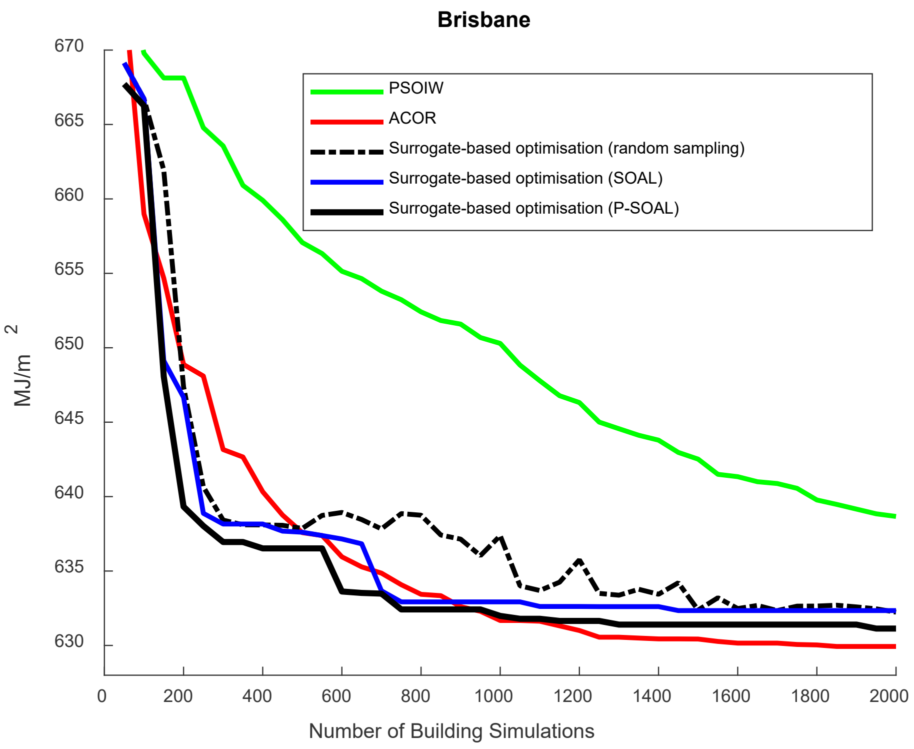

Figure 7.6: Convergence curve of the optimisation results for Brisbane (Median value of fifteen runs)

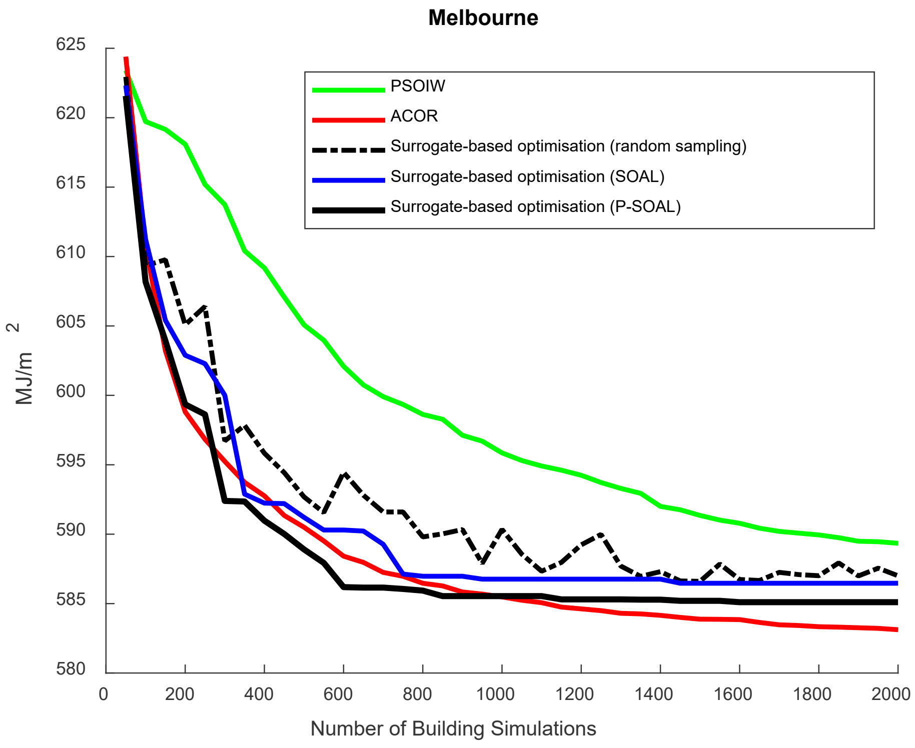

Figure 7.7: Convergence curve of the optimisation results for Melbourne (Median value of fifteen runs)

Table 7.4 shows the best parameter sets among all fifteen runs after 2000 building simulations for each algorithm for Brisbane and Melbourne. For both cities, the best solutions (the bold rows in the table) were obtained by ACOR while PSOIW found the worst solution. This table shows that the optimised building orientations are approximately 11 and 43 degrees relative to North (clockwise) for Brisbane and Melbourne, respectively. For both cities, the optimised wall has the minimum solar absorptance, and the optimised roof has the maximum emissivity with minimum solar absorptance. For Melbourne, the maximum wall insulation thickness was selected by the optimisation algorithm, while the minimum insulation thickness was chosen for Brisbane. This is likely due to Brisbane's climate, where buildings frequently have a little-to-no heating loads and high internal loads in the buildings during daytime [91]. With regard to window size, the minimum value was selected for all building' faces, except for the variable x_{12} (west overhang depth) in Brisbane. The optimised value for overhang depends on city and building direction. The minimum and maximum were selected for heating and cooling set-points for all cities, respectively. This is clearly expected when thermal comfort is not considered in the objective function and only as a constraint on the allowable range of indoor temperature set points.

From an energy point-of-view, the difference between optimised objective functions obtained by ACOR and surrogate-based optimisation using active learning methods may be considered small. Despite these small differences, different sets of parameters have been obtained, which shows that the building objective function is very multi-modal.

Table 7-4: Best parameter sets of optimisation results

	Method	**Objective Function (MJ/m^2)**	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}
Brisbane	Surrogate with random sampling	630.60	0.9	0.3	1	0.3	0.5	0.5	0.5	0.5	0.29	0.25	0.34	0.65	18	27	11.09
	SOAL	630.48	0.9	0.3	1	0.3	0.5	0.5	0.5	0.66	0.35	0.39	0.36	1.50	18	27	8.78
	P-SOAL	630.18	0.9	0.3	1	0.3	0.5	0.5	0.5	0.7	0.33	0.23	0.47	1.50	18	27	10.23
	ACOR	**629.62**	**0.88**	**0.3**	**1**	**0.3**	**0.5**	**0.5**	**0.5**	**0.72**	**0.55**	**0.54**	**0.54**	**1.44**	**18**	**27**	**11.11**
	PSOIW	635.31	0.72	0.42	1	0.31	0.58	0.57	0.87	0.52	0.63	0.55	0.84	0.39	18	27	24.14
Melbourne	Surrogate with random sampling	583.89	0.9	0.3	10	0.3	0.5	0.5	0.5	0.5	0.26	0	0.54	0.89	18	27	0
	SOAL	581.73	0.9	0.3	10	0.3	0.5	0.5	0.5	0.5	0.39	0.43	0.21	0.52	18	27	45
	P-SOAL	581.53	0.9	0.3	10	0.3	0.5	0.5	0.5	0.5	0.38	0.39	0.26	0.54	18	27	40.83
	ACOR	**580.51**	**0.89**	**0.31**	**10**	**0.31**	**0.5**	**0.5**	**0.51**	**0.51**	**0.53**	**0.52**	**0.24**	**0.54**	**18**	**27**	**42.76**
	PSOIW	585.64	0.79	0.3	8	0.47	0.51	0.5	0.52	0.51	0.37	0.06	0.56	0.82	18	27	9.23

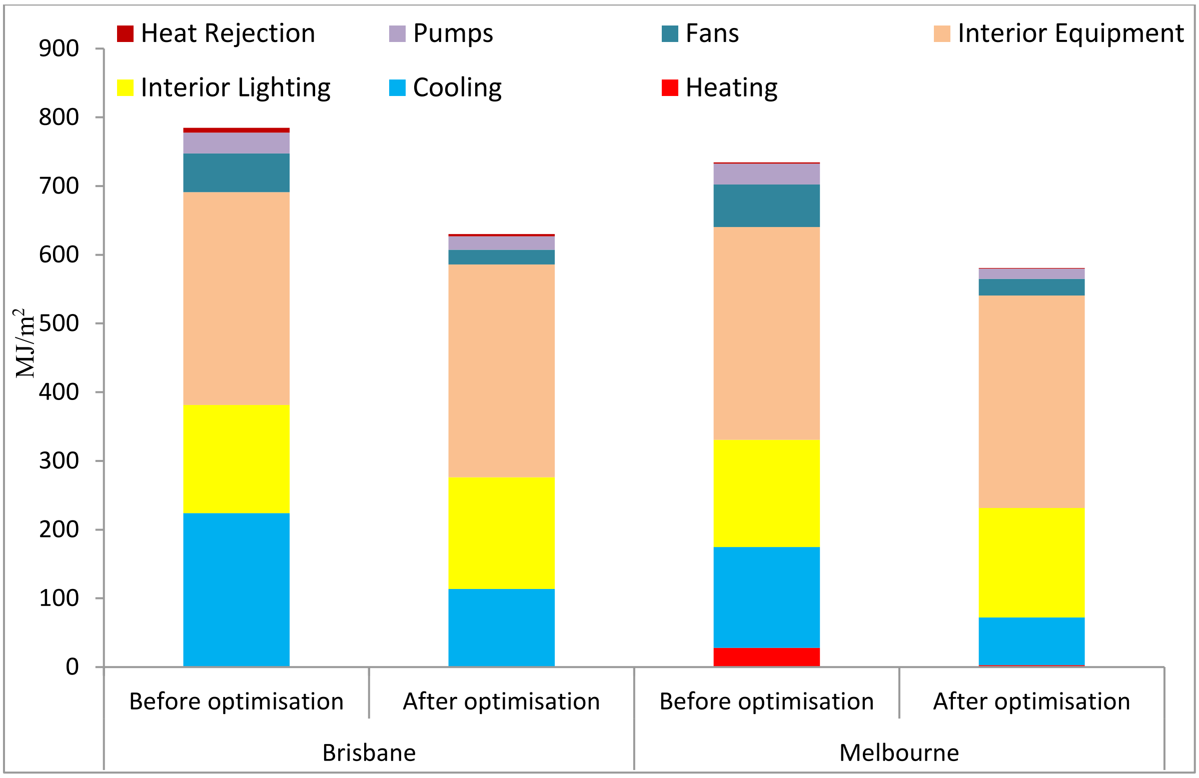

Figure 7.8: Breakdown of energy consumption before and after optimisation

Figure 7.8 shows the building annual energy consumption and the breakdown of energy consumption before and after optimisation for Brisbane and Melbourne. After applying the optimisation method, the annual energy consumption was reduced by 19.7% and 20.9% for Brisbane and Melbourne, respectively. Comparison of energy breakdown between non-optimised and optimised building shows that optimisation has significantly reduced the fan (approximately 61%) and cooling loads (approximately 20%) for both cities. The fan energy consumptions were reduced $34.6\ \mathrm{MJ/m^2}$ and $37.9\ \mathrm{MJ/m^2}$ for Brisbane and Melbourne, respectively. The cooling loads dropped $109.8\ \mathrm{MJ/m^2}$ and $76.8\ \mathrm{MJ/m^2}$ for Brisbane and Melbourne, respectively.

It is noteworthy that despite the use of daylighting control, lighting loads almost remain unchanged before and after optimisation. The reason is that minimising the cooling and lighting loads are conflicting objectives, therefore, the optimisation algorithm prioritises reduction of the cooling loads. Since the optimisation seeks the best balance between the various building loads, it is highly likely that an attempt to further decrease the lighting or cooling load would lead to a corresponding increase of equal or greater magnitude in the other.

7.6 Conclusion

In this chapter, a new simulation-based optimisation method using active learning, called SOAL, was developed and compared with simulation-based optimisation (software in the loop) and the surrogate-based optimisation method using random sampling. For the simulation-based optimisation, two PSOIW and ACOR algorithms were used. Results showed that proposed optimisation methods based on active learning could significantly improve the performance of the surrogate-based optimisation method. Importantly, in single objective optimisation problems, the proposed method not only is a competitive method to the simulation-based optimisation method using ACOR, but also could find higher quality solutions (fairly close to the final solutions) at the early optimisation stages. This demonstrates the potential of active learning surrogate-based optimisation methods in the building design phase.

All optimisation methods were applied to optimise fifteen variables in a typical, medium-size commercial building in Brisbane and Melbourne. Results showed that after applying optimisation methods, approximately 20% energy savings were achieved for both cities. A comparison of energy breakdown between optimised and non-optimised building showed that cooling load and fan energy consumption experienced the largest energy reductions for both cities.

Chapter 8: Conclusions and Future Work

Building optimisation problems are time-consuming and complex due to multi-modal and nonlinear behaviour of building thermal performance, discontinuities in the optimisation variables (e.g. window type), uncertainty in building design parameters (e.g. alterations in building operating conditions) and discontinuities in the output of building simulation software (e.g. EnergyPlus). This high computational cost remains a key barrier to practical use of optimisation methods as a building design tool. Generally, BOPs can be categorised into two main groups: simulation-based optimisation (software-in-the-loop method) and surrogate-based optimisation methods. In this thesis, new methods were developed to improve the performance of both methods.

The first contribution of this research was to significantly improve the performance of simulation-based optimisation methods. This was accomplished by the development of two optimisation algorithms. In Chapter 4:, the ACOR algorithm was developed for BOPs with continuous variables, and in Chapter 5:, the ACOMV-M algorithm was developed for BOPs with mixed variables. Results demonstrated that both algorithms are more efficient than current building optimisation algorithms in terms of optimality, consistency, and computational cost.

The second contribution of this research was the development of a new methodology to address uncertainty of building simulation inputs during the optimisation process and to select an appropriate robust design. This was accomplished by development of a multi-objective scenario-based optimisation solved by the ACOMV-M algorithm

(Chapter 6). Results demonstrated the capability of proposed uncertainty methodology to find a robust design.

The third contribution of this research was the development of a new methodology for surrogate model-based optimisation methods (Chapter 7). This was accomplished by the development of a new sample selection method to intelligently select samples for the surrogate model construction and development of a new surrogate model-based optimisation method, based on multiple surrogate models in the optimisation loop.

Building optimisation methods can be used to effectively find the optimal value of design variables within acceptable ranges defined by designers. These methods can significantly improve the drawbacks of conventional methods such as parametric analysis which often lead to a partial improvement due to complex and non-linear interactions of design variables. The methods developed in this thesis can be used by building designers to design energy-optimised buildings as they have shown to facilitate the solving of BOPs and improve the state-of-the art in terms of optimality, speed, and consistency of the optimised results. They are expected to aid building designers in meeting energy efficiency requirements in building codes. According to research findings, applying optimisation methods to typical commercial buildings in Australia showed that:

- Optimisation can significantly reduce energy consumption of commercial buildings which are fairly robust to errors in assumptions on internal loads
- In all the locations considered in this thesis, the building envelope shape and the level of thermal insulation are strongly dependent on building internal loads so that an optimised design may not require insulation at all

- Multi-objective scenario-based optimisation method can provide a design with higher robustness to poorly known (or time-varying) building simulation assumptions (e.g. changes in internal loads)

A number of areas of future work are recommended:

- In this thesis, it was demonstrated that ACO algorithms are highly capable for BOPs. Future studies could hybridise the ACO algorithm with other algorithms (e.g. hybrid ACO and a local search algorithm) to improve the performance of ACO-based algorithms.
- This thesis mainly focused on single-objective building optimisation problems (i.e. energy consumption) while other objectives have been not been considered (e.g. thermal comfort, cost or peak demand), which will be the subject of future studies.
- This research demonstrated that surrogate-based optimisation methods are very promising to reduce computational cost of BOPs. However, building optimisation using surrogate models are still in the early stages of development and new sample selection methods could be devised to further improve the performance of surrogate-based optimisation methods. Some studies have reported the superiority of surrogate-based optimisation methods over software-in-the-loop in terms of convergence speed in the multi-objective optimisation problems [19]. It would be expected that the proposed surrogate-based optimisation method can show better performance than both conventional surrogate-based optimisation methods and software-in-the-loop in multi-objective optimisation problems as well.

- While ANNs were successfully employed in this research, other machine learning methods (e.g. Support Vector Regression) may lead to improved performance or provide other information (e.g. uncertainty estimates) that would be advantageous for sample selection. An avenue of future work is therefore to investigate the use of other machine-learning models as surrogates.
- This research is the first work that has developed a smart sampling method for BOPs using surrogate model-based optimisation methods. Future studies could develop other sample selection methods for BOPs.
- This research investigated the uncertainty in building simulation inputs (e.g. occupancy) while other types of uncertainty such as uncertainty in thermo-physical properties of constructional materials or weather data have not been considered; these should be the subject of future studies.
- In this research, SWM was used to find Pareto optimal front by scalarization of the multi-objective optimisation problem to a set of single optimisation problems with different weights. Optimal solutions of this set of problems identify samples of the Pareto front. Other existing multi-objective optimisation algorithms seek to evolve populations to provide improved estimates of the Pareto front (e.g. NSGA-II, Strength Pareto Evolutionary Algorithm). However, the exploration of these other evolutionary approaches is left to future work.
- This thesis mainly focused on optimisation of commercial buildings in Australia. This research could be extended for optimisation of residential buildings.

Bibliography

1. UNEP-, *Buildings and Climate Change Summary for Decision-Makers*. 2009: France, available at :http://www.unep.org/sbci/pdfs/SBCI-BCCSummary.pdf.
2. *National Energy Productivity Plan 2015–2030: Annual Report*. Commonwealth of Australia, 2016.
3. *National Energy Productivity Plan 2015–2030*. Commonwealth of Australia, 2015.
4. Wang, W., H. Rivard, and R. Zmeureanu, *An object-oriented framework for simulation-based green building design optimization with genetic algorithms.* Advanced Engineering Informatics, 2005. **19**(1): p. 5-23.
5. Fesanghary, M., S. Asadi, and Z.W. Geem, *Design of low-emission and energy-efficient residential buildings using a multi-objective optimization algorithm.* Building and Environment, 2012. **49**: p. 245-250.
6. Tuhus-Dubrow, D. and M. Krarti, *Genetic-algorithm based approach to optimize building envelope design for residential buildings.* Building and Environment, 2010. **45**(7): p. 1574-1581.
7. Rapone, G. and O. Saro, *Optimisation of curtain wall façades for office buildings by means of PSO algorithm.* Energy and Buildings, 2012. **45**(0): p. 189-196.
8. Znouda, E., N. Ghrab-Morcos, and A. Hadj-Alouane, *Optimization of Mediterranean building design using genetic algorithms.* Energy and Buildings, 2007. **39**(2): p. 148-153.
9. Bichiou, Y. and M. Krarti, *Optimization of envelope and HVAC systems selection for residential buildings.* Energy and Buildings, 2011. **43**(12): p. 3373-3382.
10. Bouchlaghem, N., *Optimising the design of building envelopes for thermal performance.* Automation in Construction, 2000. **10**(1): p. 101-112.
11. Nielsen, T.R., *Optimization of buildings with respect to energy and indoor environment*. 2002, Technical university of Denmark.
12. Ferrara, M., et al., *A simulation-based optimization method for cost-optimal analysis of nearly Zero Energy Buildings.* Energy and Buildings, 2014. **84**: p. 442-457.

13. Bojić, M., M. Miletić, and L. Bojić, *Optimization of thermal insulation to achieve energy savings in low energy house (refurbishment).* Energy Conversion and Management, 2014. **84**: p. 681-690.

14. Ihm, P. and M. Krarti, *Design optimization of energy efficient residential buildings in Tunisia.* Building and Environment, 2012. **58**: p. 81-90.

15. Nguyen, A.-T., S. Reiter, and P. Rigo, *A review on simulation-based optimization methods applied to building performance analysis.* Applied Energy, 2014. **113**: p. 1043-1058.

16. Attia, S., et al., *Computational optimization for Zero Energy Buildings Design: Interview results with twenty-eight International experts*. 2013.

17. Attia, S., et al., *Assessing gaps and needs for integrating building performance optimization tools in net zero energy buildings design.* Energy and Buildings, 2013. **60**: p. 110-124.

18. Martí, R., M.G.C. Resende, and C.C. Ribeiro, *Multi-start methods for combinatorial optimization.* European Journal of Operational Research, 2013. **226**(1): p. 1-8.

19. Magnier, L. and F. Haghighat, *Multiobjective optimization of building design using TRNSYS simulations, genetic algorithm, and Artificial Neural Network.* Building and Environment, 2010. **45**(3): p. 739-746.

20. Ascione, F., et al., *Artificial neural networks to predict energy performance and retrofit scenarios for any member of a building category: A novel approach.* Energy, 2017. **118**: p. 999-1017.

21. Machairas, V., A. Tsangrassoulis, and K. Axarli, *Algorithms for optimization of building design: A review.* Renewable and Sustainable Energy Reviews, 2014. **31**: p. 101-112.

22. Khan, M.A.I., C.J. Noakes, and V.V. Toropov, *Multi-objective Optimization of the Ventilation System Design in a Two-bed Hospital Ward with an Emphasis on Infection Control*, in *First Building Simulation and Optimization Conference* 2012: Loughborough, UK.

23. Jérôme., C.-B., *A methodology for the optimization of building energy, thermal, and visual performance* in *Building, Civil and Environmental Engineering*. 2009, Concordia University.

24. Wetter, M. and J. Wright, *A comparison of deterministic and probabilistic optimization algorithms for nonsmooth simulation-based optimization.* Building and Environment, 2004. **39**(8): p. 989-999.

25. Wetter, M. and E. Polak, *A convergent optimization method using pattern search algorithms with adaptive precision simulation*, in *Eighth International IBPSA Conference*. 2003: Eindhoven, Netherlands

26. Baños, R., et al., *Optimization methods applied to renewable and sustainable energy: A review.* Renewable and Sustainable Energy Reviews, 2011. **15**(4): p. 1753-1766.

27. Evins, R., *A review of computational optimisation methods applied to sustainable building design.* Renewable and Sustainable Energy Reviews, 2013. **22**: p. 230-245.

28. Peippo, K., P.D. Lund, and E. Vartiainen, *Multivariate optimization of design trade-offs for solar low energy buildings.* Energy and Buildings, 1999. **29**(2): p. 189-205.

29. Michalek, J., R. Choudhary, and P. Papalambros, *Architectural layout design optimization.* Engineering Optimization, 2002. **34**(5): p. 461-484.

30. K. Shea, A.S., G. Antonuntto, *Multicriteria Optimization of Paneled Building Envelopes Using Ant Colony Optimization.* Intelligent Computing in Engineering and Architecture 2006: p. 627–636.

31. Chantrelle, F.P., et al., *Development of a multicriteria tool for optimizing the renovation of buildings.* Applied Energy, 2011. **88**(4): p. 1386-1394.

32. Ochoa, C.E., et al., *Considerations on design optimization criteria for windows providing low energy consumption and high visual comfort.* Applied Energy, 2012. **95**: p. 238-245.

33. Méndez Echenagucia, T., et al., *The early design stage of a building envelope: Multi-objective search through heating, cooling and lighting energy performance analysis.* Applied Energy, 2015. **154**: p. 577-591.

34. Delgarm, N., et al., *Multi-objective optimization of the building energy performance: A simulation-based approach by means of particle swarm optimization (PSO).* Applied Energy, 2016. **170**: p. 293-303.

35. Lin, Y.-H., et al., *Design optimization of office building envelope configurations for energy conservation.* Applied Energy, 2016. **171**: p. 336-346.

36. Wolpert, D.H. and W.G. Macready, *No free lunch theorems for optimization.* Evolutionary Computation, IEEE Transactions on, 1997. **1**(1): p. 67-82.

37. Wetter, M. and J. Wright, *Comparison of a generalized pattern search and a genetic algorithm optimization method*, in *8th IBPSA conference*. 2003: Eindhoven, Netherlands. p. 1401–8.

38. Zhou, G., et al., *Integration of an internal optimization module within EnergyPlus*. 2003.

39. Mahdavi, A. and P. Mahattanatawe, *Enclosure systems design and control support via dynamic simulation-assisted optimization*, in *Eighth International IBPSA Conference*. 2003: Eindhoven, Netherlands.

40. Wright, J. and A. Alajmi, *The robustness of genetic algorithms in solving unconstrained building optimisation problems*. 2005.

41. Hamdy, M., M. Palonen, and A. Hasan, *Implementation of pareto-archive NSGA-II algorithms to a nearly-zero-energy building optimisation problem*, in *First building simulation and optimization conference*. 2012: Loughborough, UK.

42. Hamdy, M., A.-T. Nguyen, and J.L.M. Hensen, *A performance comparison of multi-objective optimization algorithms for solving nearly-zero-energy-building design problems.* Energy and Buildings, 2016. **121**: p. 57-71.

43. Bucking, S., R. Zmeureanu, and A. Athienitis, *An information driven hybrid evolutionary algorithm for optimal design of a Net Zero Energy House.* Solar Energy, 2013. **96**(0): p. 128-139.

44. Kämpf, J.H., M. Wetter, and D. Robinson, *A comparison of global optimization algorithms with standard benchmark functions and real-world applications using EnergyPlus.* Journal of Building Performance Simulation, 2010. **3**(2): p. 103-120.

45. Ramallo-González, A.P. and D.A. Coley, *Using self-adaptive optimisation methods to perform sequential optimisation for low-energy building design.* Energy and Buildings, 2014. **81**: p. 18-29.

46. Bornatico, R., et al., *Optimal sizing of a solar thermal building installation using particle swarm optimization.* Energy, 2012. **41**(1): p. 31-37.

47. Junghans, L. and N. Darde, *Hybrid single objective genetic algorithm coupled with the simulated annealing optimization method for building optimization.* Energy and Buildings, 2015. **86**: p. 651-662.

48. Futrell, B.J., E.C. Ozelkan, and D. Brentrup, *Optimizing complex building design for annual daylighting performance and evaluation of optimization algorithms.* Energy and Buildings, 2015. **92**: p. 234-245.

49. Bambrook, S.M., A.B. Sproul, and D. Jacob, *Design optimisation for a low energy home in Sydney.* Energy and Buildings, 2011. **43**(7): p. 1702-1711.

50. Wetter, M., *GenOpt, generic optimization program version 3.1*, s.r.g. Building technologies program, Lawrence Berkeley National Laboratory, Editor. 2008.

51. Coffey, B., et al., *A software framework for model predictive control with GenOpt.* Energy and Buildings, 2010. **42**(7): p. 1084-1092.

52. Congradac, V. and F. Kulic, *HVAC system optimization with CO2 concentration control using genetic algorithms.* Energy and Buildings, 2009. **41**(5): p. 571-577.

53. Djuric, N., G. Huang, and V. Novakovic, *Data fusion heat pump performance estimation.* Energy and Buildings, 2011. **43**(2-3): p. 621-630.

54. Djuric, N., et al., *Optimization of energy consumption in buildings with hydronic heating systems considering thermal comfort by use of computer-based tools.* Energy and Buildings, 2007. **39**(4): p. 471-477.

55. Hasan, A., M. Vuolle, and K. Sirén, *Minimisation of life cycle cost of a detached house using combined simulation and optimisation.* Building and Environment, 2008. **43**(12): p. 2022-2034.

56. Park, C.-S., et al., *Real-time optimization of a double-skin façade based on lumped modeling and occupant preference.* Building and Environment, 2004. **39**(8): p. 939-948.

57. Henze, G.P., C. Felsmann, and G. Knabe, *Evaluation of optimal control for active and passive building thermal storage.* International Journal of Thermal Sciences, 2004. **43**(2): p. 173-183.

58. Griego, D., M. Krarti, and A. Hernández-Guerrero, *Optimization of energy efficiency and thermal comfort measures for residential buildings in Salamanca, Mexico.* Energy and Buildings, 2012. **54**: p. 540-549.

59. Rhodes, J.D., et al., *Using BEopt (EnergyPlus) with energy audits and surveys to predict actual residential energy usage.* Energy and Buildings, 2015. **86**: p. 808-816.

60. Palonen, M., M. Hamdy, and A. Hasan, *MOBO a new software for multi-objective building performance optimization*. 2013.

61. Korolija, I., *Heating, Ventilating and Air-conditioning System Energy Demand Coupling with Building Loads for Office Buildings*. 2011, De Montfort University, Leicester, UK.

62. Melo, A.P., et al. *Is Thermal Insulation Always Beneficial in Hot Climate?* in *14th Conference of International Building Performance Simulation Association*. 2015. Hyderabad, India.

63. Crombecq, K., E. Laermans, and T. Dhaene, *Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling.* European Journal of Operational Research, 2011. **214**(3): p. 683-696.

64. Zhou, Q., et al., *An active learning metamodeling approach by sequentially exploiting difference information from variable-fidelity models.* Advanced Engineering Informatics, 2016. **30**(3): p. 283-297.

65. Kecman, V., *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models*. 2001: MIT Press. 608.

66. Cortes, C. and V. Vapnik, *Support-vector networks.* Machine Learning, 1995. **20**(3): p. 273-297.

67. Haykin, S.S., *Neural networks and learning machines*. Vol. 3. 2009: Pearson Upper Saddle River, NJ, USA:.

68. Khayatian, F., L. Sarto, and G. Dall'O', *Application of neural networks for evaluating energy performance certificates of residential buildings.* Energy and Buildings, 2016. **125**: p. 45-54.

69. Melo, A.P., et al., *Development of surrogate models using artificial neural network for building shell energy labelling.* Energy Policy, 2014. **69**: p. 457-466.

70. Naji, S., et al., *Estimating building energy consumption using extreme learning machine method.* Energy, 2016. **97**: p. 506-516.

71. Buratti, C., M. Barbanera, and D. Palladino, *An original tool for checking energy performance and certification of buildings by means of Artificial Neural Networks.* Applied Energy, 2014. **120**: p. 125-132.

72. Paudel, S., et al., *Pseudo dynamic transitional modeling of building heating energy demand using artificial neural network.* Energy and Buildings, 2014. **70**: p. 81-93.

73. Kalogirou, S., M. Eftekhari, and L. Marjanovic-Halburd, *Estimation of the Daily Heating and Cooling Loads Using Artificial Neural Networks*. 2017.

74. Neto, A.H. and F.A.S. Fiorelli, *Comparison between detailed model simulation and artificial neural network for forecasting building energy consumption.* Energy and Buildings, 2008. **40**(12): p. 2169-2176.

75. Dong, B., C. Cao, and S.E. Lee, *Applying support vector machines to predict building energy consumption in tropical region.* Energy and Buildings, 2005. **37**(5): p. 545-553.

76. Li, Q., et al., *Predicting hourly cooling load in the building: A comparison of support vector machine and different artificial neural networks.* Energy Conversion and Management, 2009. **50**(1): p. 90-96.

77. Melo, A.P., et al., *A novel surrogate model to support building energy labelling system: A new approach to assess cooling energy demand in commercial buildings.* Energy and Buildings, 2016. **131**: p. 233-247.

78. Romero, D., J. Rincón, and N. Almao, *Optimization of the thermal behavior of tropical buildings*, in *Seventh International IBPSA Conference*. August 13-15, 2001.

79. J, C., *A methodology for the optimization of building energy, thermal, and visual performance*. 2008, Concordia University, Canada.

80. E. Tresidder, Y.Z., I. Alexander, .J. Forrester, *OPTIMISATION OF LOW-ENERGY BUILDING DESIGN USING SURROGATE MODELS*, in *12th Conference of International Building Performance Simulation Association*. 2011: Sydney.

81. E. Tresidder, Y.Z., I. Alexander, .J. Forrester, *Acceleration of building design optimisation through the use of Kriging surrogate models*, in *Proceedings of building simulation and optimization* 2012: Loughborough University, Loughborough, Leicestershire

82. Eisenhower, B., et al., *A methodology for meta-model based optimization in building energy models.* Energy and Buildings, 2012. **47**: p. 292-301.

83. Gossard, D., B. Lartigue, and F. Thellier, *Multi-objective optimization of a building envelope for thermal performance using genetic algorithms and artificial neural network.* Energy and Buildings, 2013. **67**: p. 253-260.

84. Gengembre, E., et al., *A Kriging constrained efficient global optimization approach applied to low-energy building design problems.* Inverse Problems in Science and Engineering, 2012. **20**(7): p. 1101-1114.

85. Asadi, E., et al., *Multi-objective optimization for building retrofit: A model using genetic algorithm and artificial neural network and an application.* Energy and Buildings, 2014. **81**: p. 444-456.

86. Shan, S. and G.G. Wang, *Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions.* Structural and Multidisciplinary Optimization, 2010. **41**(2): p. 219-241.

87. Settles, B., *Active learning literature survey*. 2010: University of Wisconsin–Madison.

88. Krogh;, A. and J. Vedelsby, *Neural network ensembles, cross validation, and active learning,* Advances in neural information processing systems, 1995. **7**: p. 231-238.

89. RayChaudhuri, T. and L.G.C. Hamey. *Minimisation of data collection by active learning*. in *Neural Networks, 1995. Proceedings., IEEE International Conference on*. 1995.

90. Burbidge, R., J.J. Rowland, and R.D. King, *Active Learning for Regression Based on Query by Committee*, in *Intelligent Data Engineering and Automated Learning - IDEAL 2007: 8th International Conference, Birmingham, UK, December 16-19, 2007. Proceedings*, H. Yin, et al., Editors. 2007, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 209-218.

91. Cohn, D.A., Z. Ghahramani, and M.I. Jordan, *Active learning with statistical models.* J. Artif. Int. Res., 1996. **4**(1): p. 129-145.

92. Yu, H. and S. Kim. *Passive Sampling for Regression*. in *2010 IEEE International Conference on Data Mining*. 2010.

93. Douak, F., F. Melgani, and N. Benoudjit, *Kernel ridge regression with active learning for wind speed prediction.* Applied Energy, 2013. **103**: p. 328-340.

94. Zhao, X., et al. *Selective sampling using active learning for short-term wind speed prediction*. in *2017 29th Chinese Control And Decision Conference (CCDC)*. 2017.

95. Verrelst, J., et al., *Active Learning Methods for Efficient Hybrid Biophysical Variable Retrieval.* IEEE Geoscience and Remote Sensing Letters, 2016. **13**(7): p. 1012-1016.

96. Tuia, D., et al., *A Survey of Active Learning Algorithms for Supervised Remote Sensing Image Classification.* IEEE Journal of Selected Topics in Signal Processing, 2011. **5**(3): p. 606-617.

97. Demir, B., C. Persello, and L. Bruzzone, *Batch-Mode Active-Learning Methods for the Interactive Classification of Remote Sensing Images.* IEEE Transactions on Geoscience and Remote Sensing, 2011. **49**(3): p. 1014-1031.

98. Patra, S. and L. Bruzzone, *A cluster-assumption based batch mode active learning technique.* Pattern Recognition Letters, 2012. **33**(9): p. 1042-1048.

99. Hopfe, C.J. and J.L.M. Hensen, *Uncertainty analysis in building performance simulation for design support.* Energy and Buildings, 2011. **43**(10): p. 2798-2805.

100. Hopfe, C.J., et al., *Robust multi-criteria design optimisation in building design*, in *First Building Simulation and Optimization Conference* 2012: Loughborough, UK

101. Jia, M., R.S. Srinivasan, and A.A. Raheem, *From occupancy to occupant behavior: An analytical survey of data acquisition technologies, modeling methodologies and simulation coupling mechanisms for building energy efficiency.* Renewable and Sustainable Energy Reviews, 2017. **68, Part 1**: p. 525-540.

102. Daly, D., P. Cooper, and Z. Ma, *Understanding the risks and uncertainties introduced by common assumptions in energy simulations for Australian commercial buildings.* Energy and Buildings, 2014. **75**: p. 382-393.

103. Herrando, M., et al., *Energy Performance Certification of Faculty Buildings in Spain: The gap between estimated and real energy consumption.* Energy Conversion and Management, 2016. **125**: p. 141-153.

104. Gaetani, I., P.-J. Hoes, and J.L.M. Hensen, *On the sensitivity to different aspects of occupant behaviour for selecting the appropriate modelling complexity in building performance predictions.* Journal of Building Performance Simulation, 2016: p. 1-11.

105. Rezaee, R., et al., *Assessment of uncertainty and confidence in building design exploration.* Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 2015. **29**(4): p. 429-441.

106. Menezes, A.C., et al., *Predicted vs. actual energy performance of non-domestic buildings: Using post-occupancy evaluation data to reduce the performance gap.* Applied Energy, 2012. **97**: p. 355-364.

107. Bucking, S., R. Zmeureanu, and A. Athienitis, *A methodology for identifying the influence of design variations on building energy performance.* Journal of Building Performance Simulation, 2014. **7**(6): p. 411-426.

108. Rezaee, R., et al., *A new approach to the integration of energy assessment tools in CAD for early stage of design decision-making considering uncertainty*, in *eWork and eBusiness in Architecture, Engineering and Construction*. 2014, CRC Press. p. 367-373.

109. Silva, A.S. and E. Ghisi, *Uncertainty analysis of user behaviour and physical parameters in residential building performance simulation.* Energy and Buildings, 2014. **76**: p. 381-391.

110. Se-Hoon Hyun, Cheol-Soo Park, and Godfried Augenbroe. *Uncertainty and sensitivity analysis of natural ventilation in high-rise apartment buildings*. in *Building Simulation*. 2007. Beijing, China

111. Heo, Y., R. Choudhary, and G.A. Augenbroe, *Calibration of building energy models for retrofit analysis under uncertainty.* Energy and Buildings, 2012. **47**: p. 550-560.

112. P. Hoes, et al., *Optimizing building designs using a robustness indicator with respect to user behavior*, in *12th Conference of International Building Performance Simulation Association*. 2011: Sydney,.

113. Scott Bucking, *Optimization under economic uncertainty using a net zero energy commercial office case study.* ASHRAE Transactions, 2016. **122**: p. 444-454.

114. Ramallo-González, A.P., T.S. Blight, and D.A. Coley, *New optimisation methodology to uncover robust low energy designs that accounts for occupant behaviour or other unknowns.* Journal of Building Engineering, 2015. **2**: p. 59-68.

115. Li, Y.F., et al., *A systematic comparison of metamodeling techniques for simulation optimization in Decision Support Systems.* Applied Soft Computing, 2010. **10**(4): p. 1257-1273.

116. Pallottino, S., G.M. Sechi, and P. Zuddas, *A DSS for water resources management under uncertainty by scenario analysis.* Environmental Modelling & Software, 2005. **20**(8): p. 1031-1042.

117. Kotireddy, R., P.-J. Hoes, and J.L.M. Hensen, *A methodology for performance robustness assessment of low-energy buildings using scenario analysis.* Applied Energy, 2018. **212**: p. 428-442.

118. *EnergyPlus Energy Simulation Software*. 2015; Available from: http://apps1.eere.energy.gov/buildings/energyplus/.

119. Wisconsin, U.o. *Transient system simulation program (TRNSYS)*. Available from: http://www.trnsys.com/.

120. Ma, Y., et al., *Comparison of Different Solar-Assisted Air Conditioning Systems for Australian Office Buildings.* Energies, 2017. **10**(10): p. 1463.

121. *Integrated Environmental Solutions <Virtual Environment>*. Available from: https://www.iesve.com/.

122. Corgnati, S.P., et al., *Reference buildings for cost optimal analysis: Method of definition and application.* Applied Energy, 2013. **102**(Supplement C): p. 983-993.

123. BRANZ, *An Assessment of the Need to Adapt Buildings for the Unavoidable Consequences of Climate Chang*. 2007, Australian Greenhouse Office, Department of the Environment and Water Resources,: Canberra

124. Guan, L., *Sensitivity of Building Zones to Potential Global Warming.* Architectural Science Review, 2009. **52**(4): p. 279-294.

125. Guan, L., *Implication of global warming on air-conditioned office buildings in Australia.* Building Research & Information, 2009. **37**(1): p. 43-54.

126. Pitt&Sherry, *The Pathway to 2020 for Low-Energy, Low-Carbon Buildings in Australia: Indicative Stringency Study*. 2010, Department of Climate Change and Energy Efficiency, : Canberra.

127. Guan, L., *Sensitivity of building cooling loads to future weather predictions.* Architectural Science Review, 2011. **54**(3): p. 178-191.

128. Samarakoon, E. and V. Soebarto, *Testing the Sensitivity of User Patterns in Building Energy Performance Simulation*, in *12th Conference of International Building Performance Simulation Association*. 2011: Sydney.

129. National Australian Built Environment Rating Scheme, *NABERS Energy Guide to Building Energy Estimation*. 2011.

130. Guan, L., *Energy use, indoor temperature and possible adaptation strategies for air-conditioned office buildings in face of global warming.* Building and Environment, 2012. **55**: p. 8-19.

131. Daly, D., P. Cooper, and Z. Ma, *Implications of global warming for commercial building retrofitting in Australian cities.* Building and Environment, 2014. **74**: p. 86-95.

132. Ma, Y. and L. Guan, *Performance Analysis of Solar Desiccant-Evaporative Cooling for a Commercial Building under Different Australian Climates.* Procedia Engineering, 2015. **121**: p. 528-535.

133. Ma, Y., et al., *Parametric Analysis of Design Parameter Effects on the Performance of a Solar Desiccant Evaporative Cooling System in Brisbane, Australia.* Energies, 2017. **10**(7): p. 849.

134. Board, A.B.C., *ABCB Energy Modelling of Office Buildings For Climate Zoning (Class 5 Climate Zoning Consultancy) Stages 1, 2 & 3*. 2002, Australian Building Codes Board.

135. Board, A.B.C., *ABCB Energy Modelling of Office Buildings For Climate Zoning (Class 5 Climate Zoning Consultancy) Stages 4 & 5*. 2002, Australian Building Codes Board.

136. Bannister, P., *Australian Building Codes Board: Class 5 Benchmarking*. 2004, Exergy Australia Pty.

137. Socha, K. and M. Dorigo, *Ant colony optimization for continuous domains.* European Journal of Operational Research, 2008. **185**(3): p. 1155-1173.

138. Zhang, B., et al., *Application of homogenous continuous Ant Colony Optimization algorithm to inverse problem of one-dimensional coupled radiation and conduction heat transfer.* International Journal of Heat and Mass Transfer, 2013. **66**(0): p. 507-516.

139. Bonabeau, E., M. Dorigo, and G. Theraulaz, *Swarm intelligence: from natural to artificial systems*. 1999: Oxford University Press, Inc. 307.

140. Niknam, T. and B. Amiri, *An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis.* Applied Soft Computing, 2010. **10**(1): p. 183-197.

141. Niu, D., Y. Wang, and D.D. Wu, *Power load forecasting using support vector machine and ant colony optimization.* Expert Systems with Applications, 2010. **37**(3): p. 2531-2539.

142. Dorigo, M., V. Maniezzo, and A. Colorni, *Ant system: optimization by a colony of cooperating agents.* Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 1996. **26**(1): p. 29-41.

143. Kennedy, J. and R. Eberhart. *Particle swarm optimization*. in *Neural Networks, 1995. Proceedings., IEEE International Conference on*. 1995.

144. Yuhui, S. and R. Eberhart. *A modified particle swarm optimizer*. in *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*. 1998.

145. Yuhui, S. and R.C. Eberhart. *Empirical study of particle swarm optimization*. in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*. 1999.

146. Hook, R. and T.A. Jeeves, *Direct search solutions of numerical and statistical problems.* Journal of the Association for Computing Machinery, 1961. **8**: p. 212–229.

147. Nelder, J.A. and R. Mead, *A Simplex Method for Function Minimization.* The Computer Journal, 1965. **7**(4): p. 308-313.

148. ABCB, *(Australian Building Codes Board), Handbook: NCC Volume One Energy Efficiency Provisions*. 2018.

149. Pitt&Sherry, *The Pathway to 2020 for Low-Energy Low-Carbon Buildings in Australia: Indicative Stringency Study*. July 2010.

150. Wetter, M., *GenOpt-Generic Optimization Program. User Manual Version 3.1.0*. 2011: Lawrence Berkeley National Laboratory, Berkeley, CA, USA.

151. Lam T. Bui, O.S., Hussein A. Abbass, *A Modified Strategy for the Constriction Factor in Particle Swarm Optimization.* Lecture Notes in Computer Science, 2007. **4828**: p. 333-344.

152. Guan, L.-S. *Will insulation always bring benefits in energy saving and thermal comfort?* in *Proceedings of the First International Conference on Sustainable Urbanization ICSU* 2010. The Hong Kong Polytechnic University, China, Hong Kong

153. Diakaki, C., et al., *A multi-objective decision model for the improvement of energy efficiency in buildings.* Energy, 2010. **35**(12): p. 5483-5496.

154. Asadi, E., et al., *Multi-objective optimization for building retrofit strategies: A model and an application.* Energy and Buildings, 2012. **44**: p. 81-87.

155. Liao, T., et al., *Ant Colony Optimization for Mixed-Variable Optimization Problems.* IEEE Transactions on Evolutionary Computation, 2014. **18**(4): p. 503-518.

156. Bamdad, K., et al., *Ant colony algorithm for building energy optimisation problems and comparison with benchmark algorithms.* Energy and Buildings, 2017. **154**(Supplement C): p. 404-414.

157. McCulloch, W. and W. Pitts, *A logical calculus of the ideas immanent in nervous activity.* The bulletin of mathematical biophysics, 1943. **5**(4): p. 115-133.

158. Mba, L., P. Meukam, and A. Kemajou, *Application of artificial neural network for predicting hourly indoor air temperature and relative humidity in modern building in humid region.* Energy and Buildings, 2016. **121**(Supplement C): p. 32-42.

159. Hornik, K., M. Stinchcombe, and H. White, *Multilayer feedforward networks are universal approximators.* Neural Networks, 1989. **2**(5): p. 359-366.

160. Csáji, B.C., *Approximation with Artificial Neural Networks*, in *Faculty of Sciences*. 2001, Eötvös Loránd University: Hungary.

161. Kůrková, V., *Kolmogorov's theorem and multilayer neural networks.* Neural Networks, 1992. **5**(3): p. 501-506.

162. Levenberg, K., *A METHOD FOR THE SOLUTION OF CERTAIN NON-LINEAR PROBLEMS IN LEAST SQUARES.* Quarterly of Applied Mathematics, 1944. **2**(2): p. 164-168.

163. Marquardt, D., *An Algorithm for Least-Squares Estimation of Nonlinear Parameters.* Journal of the Society for Industrial and Applied Mathematics, 1963. **11**(2): p. 431-441.

164. Haykin, S., *Neural Networks: A Comprehensive Foundation*. 1998: Prentice Hall PTR. 842.

165. Demuth, H., M. Beale, and M. Hagan, *Neural Network Toolbox 6 for Use's Guide*. 2006: The MathWorks, Inc.

166. Yu, H. and B.M. Wilamowski, *The Industrial Electronics Intelligent Systems, Levenberg–Marquardt Training*. 2011: CRC Press.

167. Meckesheimer, M., et al., *Computationally Inexpensive Metamodel Assessment Strategies.* AIAA Journal, 2002. **40**(10): p. 2053-2060.

168. Chou, J.-S. and D.-K. Bui, *Modeling heating and cooling loads by artificial intelligence for energy-efficient building design.* Energy and Buildings, 2014. **82**: p. 437-446.

169. Kohavi, R., *A study of cross-validation and bootstrap for accuracy estimation and model selection*, in *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2*. 1995, Morgan Kaufmann Publishers Inc.: Montreal, Quebec, Canada. p. 1137-1143.

170. McKay, M.D., R.J. Beckman, and W.J. Conover, *Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code.* Technometrics, 1979. **21**(2): p. 239-245.